

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Monitor biosignálů pro freediving
Biosignal Monitor for Freediving

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Zadání bakalářské práce

Student: **Josef Paleček**
Studijní program: B2649 Elektrotechnika
Studijní obor: 3901R039 Biomedicínský technik
Téma: Monitor biosignálů pro freediving
Biosignal Monitor for Freediving

Zásady pro vypracování:

1. Analýza problematiky freedivingu a výběr monitorovaných biosignálů s ohledem na požadavky freediverů.
2. Návrh a realizace zařízení pro monitorování vybraných biologických signálů.
3. Realizace software pro vizualizaci měřených dat.
4. Provedení testů realizovaných zařízení.
5. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

1. SIPPERLY, D., MAAS, T. *Freedive : BlueWater Freedivers*
Ventura: California (USA), 1998. 152 s. ISBN 0-9644966-1-5.
2. Guang-Zhong Yang (Ed.). *Body Sensor Network* London (UK): Springer-Verlag, 2006. 493 s.
ISBN-13: 978-1-84628-272-0.
3. WEBSTER, John G. *Medical instrumentation : application and design*. Hoboken (USA): Wiley, 1998.
691 s. ISBN 0-471-15368-0.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Martin Černý**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012

doc. Ing. Jiří Koziolek, Ph.D.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

„Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.“

A handwritten signature in black ink, appearing to read 'J. Paleček', written in a cursive style.

Josef Paleček

Poděkování

Rád bych poděkoval svému vedoucímu bakalářské práce Ing. Martinovi Černému, Ph.D. za čas, který mi věnoval na konzultacích, trpělivost a za připomínky a rady, kterými přispěl k vypracování této práce.

Dále děkuji za pomoc při realizaci zařízení firmě Mentos a 7Marsyas.

Zvláštní poděkování patří Davidu Čáňmu za odborné konzultace problematiky freedivingu.

Abstrakt

Tato bakalářská práce se zabývá měřením biosignálů ve freedivingu s ohledem na požadavky freedivera. Úkolem této práce byl návrh a realizace přístroje pro měření EKG a tlaku v hydrostatickém prostředí. Práce zahrnuje výběr pouzdra, tlakového senzoru a vytvoření komunikace mezi tlakovým senzorem a holterem. Je zde definováno vytvoření uživatelského prostředí k vizualizaci naměřených hodnot a testování funkčnosti senzoru. V závěru byla shrnuta celá bakalářská práce.

Klíčová slova:

freediving, holter, tlakové pouzdro, tlakový senzor, mikrokontrolér, sběrnice SPI, Matlab

Abstract

This thesis deals with the measurement of biosignals in freediving with consideration on requirements of freedivers. The target of this work was the project and implementation device for measuring EKG and pressure in hydrostatic environment. The work includes a selection of case, pressure sensor and creating the communication between the pressure sensor and holter monitor. Finally, there is a description of creation a user interface for visualization of measured values and testing the functionality of the sensor. In the conclusion the entire thesis was summed up.

Key words:

freediving, holter, pressure case, pressure sensor, microcontroller, SPI bus, Matlab

Obsah

1	Úvod.....	1
2	Teoretická část	2
2.1	Freediving	2
2.1.1	Historie.....	2
2.1.2	Vybavení pro freediving	2
2.1.3	Disciplíny freedivingu.....	3
2.1.4	Fyziologické změny a nebezpečí.....	4
2.2	Srdce a jeho činnost	5
2.2.1	Anatomie srdce.....	5
2.2.2	Vznik a šíření srdečního vzruchu	6
2.2.3	Elektrokardiografie	6
2.2.4	Elektrokardiografické svody	7
2.3	Holter	7
2.4	Snímače tlaku.....	7
2.4.1	Piezorezistivní snímače.....	8
2.5	Mikrokontrolér PIC	10
2.5.1	SPI.....	10
3	Návrh zařízení	11
3.1	Výběr holteru	11
3.2	Výběr tlakového senzoru.....	11
3.2.1	Porovnání tlakových senzorů	12
3.3	Parametry PIC24FJ64GB002	13
3.4	Senzor MS5535-30C	13
3.4.1	Popis.....	14
3.4.2	Rozměry	14
3.4.3	Konfigurace pinů.....	14
3.4.4	Obecný popis funkce senzoru	15
3.4.5	Vývojový diagram.....	17
3.5	Výběr vodotěsného pouzdra pro holter	18
3.5.1	Obal MENTES	19

4	Technická realizace	21
4.1	Propojení mikroprocesoru s tlakovým senzorem	21
4.2	Návrh a výroba plošného spoje pro tlakový senzor	21
5	Softwarová realizace	23
5.1	Obecný diagram měření tlaku a teploty	23
5.2	Komunikace mikrokontroléru a tlakového senzoru pomocí SPI sběrnice	24
5.3	Komunikace senzoru po SPI	24
5.3.1	Vytvoření hodinového impulsu MCKL	24
5.3.2	Získání kalibračních hodnot	25
5.3.3	Převod kalibračních hodnot	26
5.3.4	Získání hodnot tlaku a teploty	26
5.3.5	Výsledný výpočet tlaku a teploty	27
5.3.6	Uložení hodnot tlaku a teploty na SD kartu	28
6	Zpracování naměřených dat	30
6.1	Software pro nahrání a vykreslení naměřených dat	30
7	Testování	32
7.1	Zapojení měřicího řetězce	32
7.2	Výsledky měření	33
8	Závěr	35
9	Literatura	36
10	Přílohy	37

1 Úvod

V dnešní době se na freediving pohlíží jinak, než tomu bylo v minulosti. Zatímco v minulosti to byla hlavně činnost zajišťující obživu, dnes se lidé freedivingu věnují za účely jak rekreačními, tj. poznávání podmořského světa a odpočinek, tak i závodními. Závodní freediving je o posunování lidských možností a překonávání extrémních hloubek bez přístrojů. Většina fyziologických a patofyziologických stavů, do kterých je freediver při zvýšeném okolním tlaku vodního prostředí vystaven, není zcela známa.

Tato práce se zabývá návrhem měřicího přípravku pro monitorování biosignálů a tlaku v hydrostatickém prostředí. V práci je využito existující řešení EKG holteru bakalářské práce.

Teoretická část se zabývá rozбором tlakových senzorů a jejich popisem. Existuje mnoho druhů tlakových senzorů, a proto je třeba rozlišit jejich rozsahy měření a vybrat ten nejvhodnější. Dále je třeba charakterizovat princip činnosti EKG holteru, protože holter obsahuje mikrokontrolér, který zpracovává a ukládá naměřená data na SD kartu, je důležité znát jeho funkci. Komunikace senzoru s mikrokontrolérem je provedena skrze sériové rozhraní SPI. Všeobecný úvod do SPI komunikace je popsán v první kapitole. Obsahem kapitoly technické realizace je návrh celého měřicího řetězce. Jedna z nejdůležitějších částí je výběr pouzdra, u kterého je nutné zajištění ochrany elektroniky. V technické realizaci se řeší propojení senzoru s mikrokontrolérem. Také v této kapitole je vyřešeno uchycení tlakového senzoru k pouzdru.

Cíl práce je vytvoření komunikace mezi senzorem a mikroprocesorem po SPI sběrnici. Lze ho nalézt v kapitole softwarová realizace, kde je i popsán postup pro vytvoření spojení.

Pro zobrazení naměřených dat je vytvořeno uživatelské rozhraní, které jasně znázorňuje průběh měření.

Poslední část bakalářské práce se zabývá testováním tlakového senzoru.

2 Teoretická část

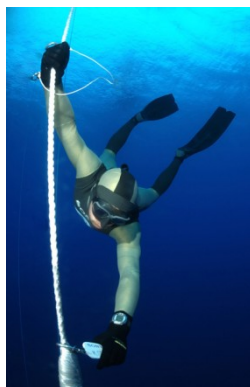
V této části je popsána teorie freedivingu, elektrokardiografie a elektronických součástek.

2.1 Freediving

2.1.1 Historie

Freediving, neboli potápění na nádech je pokročilá forma snorkelování, ať už v rekreační či závodní podobě. Již před lety se lovci perel potápěli ve světových mořích. Kolem roku 1940 se širšího podvědomí veřejnosti se freediving dostal hlavně díky filmu. Příkladem je snímek „The Big Blue“.

Dnes je již freediving provozován miliony lidí po celém světě, především při objevování krás podvodního života i prostředí na korálových útesech nebo vracích. Vše pouze za použití potápěčské masky, ploutví a dýchací trubice – snorkelu.



Obr. 1 Freediver [1]

2.1.2 Vybavení pro freediving

Tak jako u přístrojového potápění, je nutné mít také při volném potápění základní vybavení. Výstroj pro freediving není tak obsáhlá jako u přístrojového potápění. Vybavení a doplňky při potápění slouží pouze k vylepšení nebo prodloužení vlastního těla. Ve freedivingu se používají tato vybavení:

Potápěčská maska – nejdůležitější část, bez které se nedá pozorovat podvodní prostředí. Na trhu existuje široká nabídka masek různých variant a tvarů. Ne všechny jsou vhodné pro freediving. Základní funkce masky je udržet oči mimo kontakt s vodou.

Ploutve – umožňují se pohybovat pod vodou s komfortem a úsporou energie, která je pro freediving nejdůležitější. Na trhu v dnešní době je obrovský výběr různých ploutví. Různých materiálů, velikostí a tvarů, které se specializují na různé disciplíny. Základní rozdělení ploutví:

- a) Standardní – normální ploutev
Tento typ ploutví je používán u většiny vodních sportů
- b) Ploutve s dlouhým listem

Použití této ploutve umožňuje plavání na stejnou vzdálenost s menším počtem kopnutí. Průměrná délka listu je 65cm.

c) Botičky s ploutví

Úkolem botiček je co nejlépe spojit ploutev s chodidlem potápěče.

d) Monoploutev

e) Dýchací trubice /snorkel/ pro freediving

U tohoto vybavení platí čím jednodušší, tím lepší. Na trhu najdeme mnoho modifikací snorkelů.

Neopren – základním požadavkem neoprenů je tepelná izolace, prodloužení času ve vodě a ochrana povrchu těla. Na trhu se vyskytuje dvoudílný a jednodílný oblek.

Zátěžový opasek – slouží k regulaci pozitivního vztlaku vytvořeného neoprenem a vztlakem těla. Pomáhá k lepšímu vyvážení těla pod vodou při ponoru a při rozdýchávání.

Freediving počítače – počítače speciálně vytvořené pro freediving, které dodávají informace o hloubce, maximální hloubce a čase pod vodou. Mají speciální časové a hloubkové alarmy.

2.1.3 Disciplíny freedivingu

Dělíme na dvě základní kategorie.

Bazénové disciplíny:

Statická apnoe (STA) - cílem je vydržet co nejdéle na jeden nádech. Nejobvyklejší je poloha na hladině s obličejem ve vodě. Apneista musí být maximálně uvolněn. Při této disciplíně dochází ke snížení funkce metabolismu díky speciální přípravě před zadržením dechu, proto i v teplé vodě musí freediver mít neopren.

Aktuální rekord: 12:11 min

Dynamická apnoe (DYF) - se dále dělí na DYF s ploutvemi a bez ploutví. U této disciplíny je úkolem uplavat co největší vzdálenost pod vodou.

Aktuální rekord: 273m, bez ploutví: 232m

Hloubkové disciplíny:

Konstantní zátěž s ploutvemi (CWT) - Potápěč sestupuje a vystupuje pouze za pomoci ploutví. Odhození zátěže, přitahování za lano, nebo jiná dopomoc nejsou povoleny.

Aktuální rekord: 124m

Konstantní zátěž bez ploutví (CNF) - Potápěč sestupuje a vystupuje pouze za pomoci vlastních sil (bez ploutví). Odhození zátěže, přitahování za lano, nebo jiná dopomoc nejsou povoleny.

Aktuální rekord: **101m**

Free Immersion (FIM) - Zde se snaží závodník o dosažení maximální hloubky, bez pomoci jakéhokoliv propulsního vybavení /ploutve, sled apod./. Přitahování za sestupové lano je povoleno.

Aktuální rekord: 121m

Variabilní zátěž (VWT) - V této disciplíně se k sestupu používá závaží ve formě jakéhosi výtahu /tzv.sled/, který je zanechán v cílové hloubce. Potápěč vystupuje své pomoci za použití ploutví. Přitahování za lano je povoleno.

Aktuální rekord: 142m

No Limits (NLT) - Jak již název disciplíny napovídá, nejsou zde kladeny téměř žádné limity na techniku sestupu a výstupu. Obvyklá kombinace je použití sledu pro sestup a balónu naplněného vzduchem pro výstup.

Aktuální rekord 214m

[2]

2.1.4 Fyziologické změny a nebezpečí

Hypoxie – je nedostatek kyslíku v organismu. Během ponoru se kyslík v organismu průběžně spotřebovává. To nutí pracovat potápěč v hypoxickém režimu a to ovlivňuje jeho výkon. Tělo normálně pracuje, pokud úroveň kyslíku bude na min. 16%. Příznaky hypoxie mohou nastat již od 14% a ztráta vědomí (blackout) nastane při úrovni 10% kyslíku v organismu. Od tohoto okamžiku je nemožné dostatečně vyživovat buňky a může docházet k jejich odumírání.

Hyperkapnie – přebytek oxidu uhličitého ve tkáních. Příčina – nedostatečná ventilace plic, úmyslné zadržování dechu, zvýšená tvorba při práci.

Příznaky: otupělost, ztráta koncentrace až ztráta vědomí, zvýšení srdeční frekvence.

Hyperventilace – obnáší změny v CNS a omezuje krevní průtok mozku. To má za následek závratě a křeče v dolních i horních končetinách. Hyperventilace je častou příčinou nehod!

Dusíková narkóza – hloubkové opojení je spojeno se zvýšením parciálního tlaku dusíku v těle. S narůstající hloubkou a celkovým tlakem (atmosférický + hydrostatický tlak) se parciální N₂ (dusík) zvyšuje a projevuje se euforií, nesoustředěností, strachem, nesoudností atd. Při freedivingu se vyskytuje přibližně od hloubky 60-ti metrů a je umocněn špatnou viditelností, chladnou vodou, nervozitou aj.

Samby – je především známá mezi závodními freediver, stav těla, kdy organismus prochází jistým stupněm hypoxie. Není však spojen se ztrátou vědomí, projevuje se především na hladině těsně po vynoření – euforií, zmatkem potápěče, ztrátou motorické kontroly, cyanózou atd.

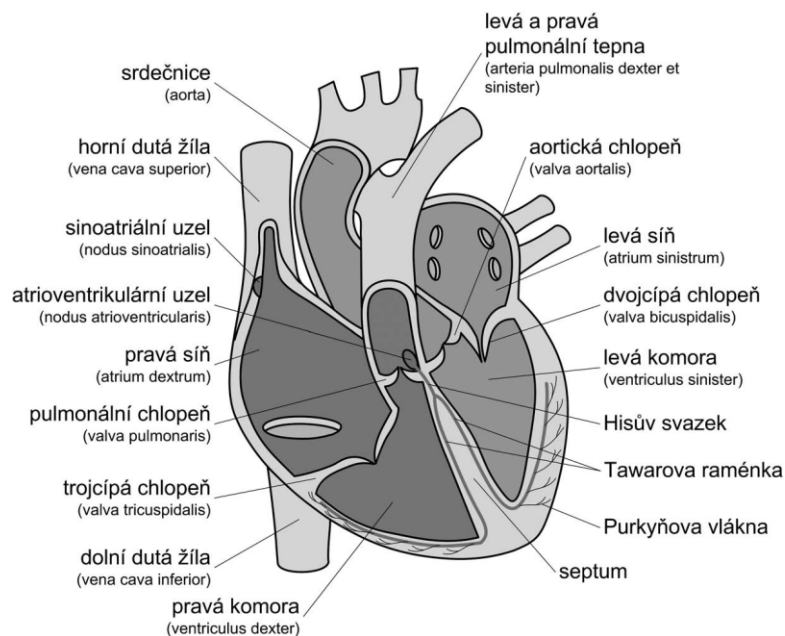
Blackout – projevy a příznaky blackoutu jsou podobné, jako u samby. Potápěč upadá do bezvědomí, není schopen se držet vlastní silou nad hladinou vody. Není výjimkou, že potápěč

po vynoření začne sám dýchat, má však jisté příznaky hypoxie a po několika sekundách upadá do bezvědomí. Vzniká u něho ztráta kontroly nad dýcháním (zástava dýchání- laryngospasmus).

2.2 Srdce a jeho činnost

2.2.1 Anatomie srdce

Jde o dutý svalový orgán tvaru pyramidy, který je uložen v hrudníku. Dvě třetiny srdce jsou uloženy v levé části hrudníku, jedna třetina v pravé části. Srdce vytváří bázi srdeční - *basis cordis* a doleva směřující hrot - *apex*. Lidské srdce váží kolem 250 – 300g, mužské více než ženské. Srdce se skládá ze dvou předsíní: pravá a levá předsíň - *atrium dextrum et sinistrum*. Na srdeční předsíně navazuje pravá a levá srdeční komora - *ventriculus dexter et sinister*. Obě předsíně vybíhají v krátká ouška – *auriculae atriales*, vpravo i vlevo. Předsíně a komory spolu komunikují otvorem – *ostium atrioventriculare*, vpravo s chlopní trojcípou - *valva tricuspidalis*, vlevo s chlopní dvojcípou - *valva bicuspidalis (mitralis)*. Tyto chlopně usměrňují průtok krve z předsíní do komor. Po obvodu jsou chlopně přirostlé k vazivovému prstenci - *anulus fibrosus* a volné konce chlopně - cípy - *cuspis*, jsou upevněny k srdeční svalovině šlašinkami - *chordae tendineae*. Z pravé komory odstupuje plicnice - *truncus pulmonalis*, z levé komory odstupuje aorta. Na začátku obou těchto velkých cév jsou chlopně poloměsíčné, semilunární - *valva aortalis* a *valva trunci pulmonalis*. Do pravé předsíně vstupuje horní a dolní dutá žila - *vena cava superior et inferior*, která přivádí odkysličenou krev z celého těla. Do levé síně ústí čtyři žíly plicní s okysličenou krví - *venae pulmonales*. *Truncus pulmonalis* přivádí z pravé komory krev odkysličenou do plic, plicní žíly přivádějí krev okysličenou z plic (zpravidla z každé plice dvě žíly) do levé předsíně.[3]



Obr. 2 Stavba srdce [4]

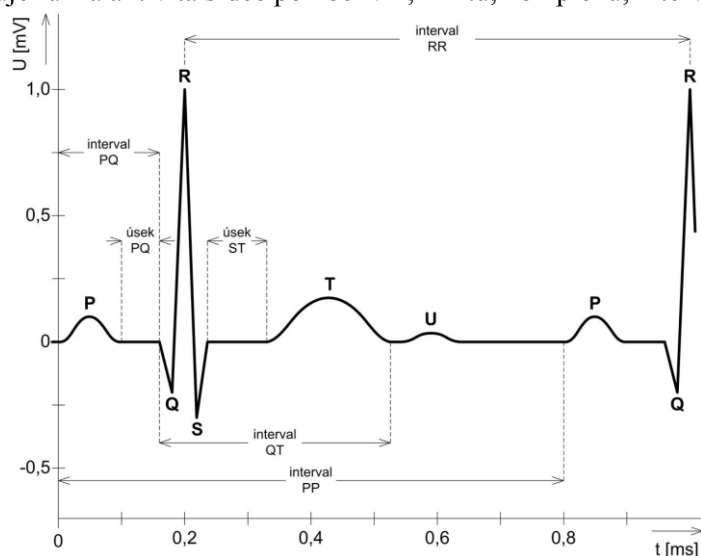
2.2.2 Vznik a šíření srdečního vzruchu

Srdeční svalovina je podobná kosternímu svalstvu, ale liší se tím, že buňky srdečního svalu jsou navzájem propojeny a to má význam pro šíření vzruchu. Srdce pracuje automaticky a rytmicky jako pulzní pumpa. Vzruch vzniká ve zvláštní srdeční svalovině (převodním systému). Tvoří ho dva uzlíky sinusový (na rozhraní vstupu horní duté žíly do pravé síně) a síňokomorový (v pravé síni na rozhraní síně a komory). Buňky převodního systému se liší od ostatních buněk srdečního svalu malým množstvím myofibril a vysokým obsahem glykogenu. Funkčně se liší tím, že jsou **nadány automacií** – schopnost rytmicky vytvářet vzruchy. Za normálních okolností vzniká vzruch v nejrychlejším centru automacie v sinusovém uzlu (60 až 70 za minutu). V síňokomorovém uzlu je frekvence podstatně nižší (40 za minutu), a proto se jako udavatel rytmu (pacemaker) neuplatňuje. Síňokomorový uzel zpomaluje vedení vzruchu přicházejícího ze sinoatriálního uzlu. Odsud se vede vzruch dále Hisovým svazkem, větví do Tawarových ramének (levé a pravé) a z nich dále pokračuje do Purkyňových vláken, které jsou poslední částí převodního systému. Vzruch putuje jenom do srdečního svalu.

[5]

2.2.3 Elektrokardiografie

Za rozvinutí a uvedení elektrokardiografie do klinické praxe dostal Willem Einthoven (1860 - 1927) Nobelovu cenu. EKG je jedno ze základních neinvazivních vyšetření myokardu využívaných v kardiologii – záznam nitrokomorového potencionálu. Signál myokardu se šíří všemi směry a když není nějak výrazně oslabeno, tak EKG lze zaznamenat (jednotkách až desítkách mV). Frekvenční rozsah 0,1 – 100 Hz, typická amplituda 1 – 10mV. Základní křivka elektrokardiogramu podle zapojení Einthovenova trojúhelníku je znázorněna na obr. 3. Na EKG záznamu se popisuje různá aktivita srdce pomocí vln, kmitů, komplexu, intervalů.



Obr. 3 : EKG křivka [4]

2.2.4 Elektrokardiografické svody

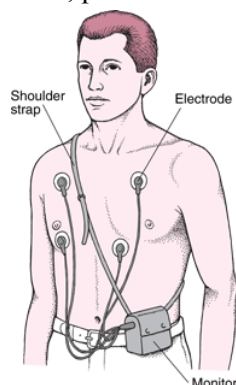
V praxi se pro snímání EKG používá zapojení 12svodů. Tyto svody jsou umístěny na různých částech těla pacienta (hrudník, končetiny).

Dvanácti svodové EKG se skládá z:

- Bipolárních Einthovenových svodů (I, II, III)
- Unipolárních Goldbergových svodů (aVR, aVL, aVF)
- Unipolárních Wilsonových svodů ($V_1 - V_6$)

2.3 Holter

EKG Holter je důležitým dlouhodobým 24-ti hodinovým kardiologickým vyšetřením, které po změření vyhodnotí lékař, a niž by pacient musel být sledován v nemocnici. Využívá se v diagnostice srdečních arytmií nebo poruch vědomí, které se během klasického krátkodobého EKG vyšetření nezjistí. Malá přenosná krabička se propojí s pacientem pomocí nalepení elektrod na hrudník. Na obr. 4, lze vidět zapojení holteru. V našem případě se bude používat standardní holter se speciálním obalem. Pomocí tohoto holteru nebude sledován 24hodinový záznam EKG, ale jen určitý časový interval, ponor freedivera.



Obr. 4 Pacient připojený k Holteru [6]

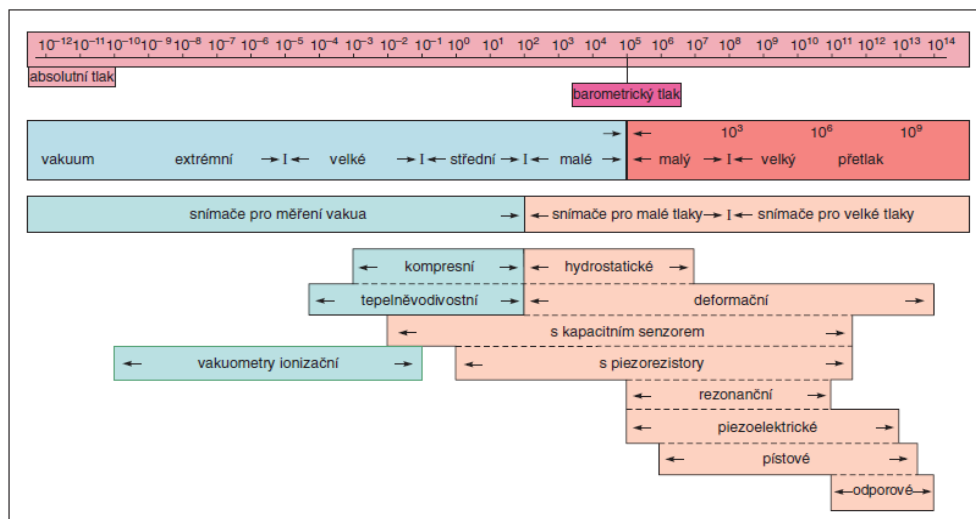
2.4 Snímače tlaku

Veličinu tlaku definujeme pomocí hydrostatického tlaku $p=h \cdot \rho \cdot g$, kde p je určen prostřednictvím hydrostatického sloupce kapaliny, výšce h a g (gravitačnímu zrychlení). Základní SI jednotka tlaku je *pascal* (Pa) ale je dovoleno se používají jednotku bar (1 bar = 100kPa). Pro měření hodnoty tlaku se používají dvě vztažné hodnoty, buď absolutní (hodnota naměřená od absolutní nuly tlaku), nebo barometrický (hodnota tlaku v okolní místa měření). Také se může měřit diferencí (rozdíl) tlaků, kde není žádná hodnota shodná s barometrickým tlakem.

Přístroje pro měření tlaku se nazývají tlakoměry. Tlakoměr (manometr) je označení pro měření přetlaku. Přístroje pro měření podtlaku se nazývají vakuometry. Pro měření rozdílu tlaku (difference) se používá diferenční tlakoměr.

Snímač tlaku převádí měřený tlak na analogový signál. Citlivá část se nazývá čidlo. Pojmy senzor a čidlo mohou mít v různých literaturách stejný význam. Jestliže snímač tlaku je řízen, nebo ho řídí mikroprocesor, mluvíme o tzv. inteligentním snímači.

Existuje mnoho typů snímačů podle principu měření, rozsahu, přesnosti, možnosti použití. Na následujícím obrázku (obr. 5) vidíme rozdělení tlakoměru podle rozsahu. [7]

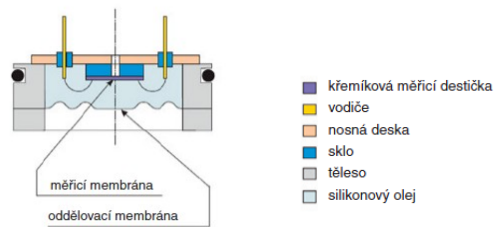


Obr. 5 Rozdělení tlakových senzorů [7]

2.4.1 Piezorezistivní snímače

Monokrystalický křemík se u piezorezistivních snímačů používá již od začátku 60. let dvacátého století. Měrný odpor krystalu je závislý na mechanickém namáhání. Měřicí soustavu snímače tvoří křemíková destička s vysokým odporem, která je mechanicky namáhána. Vodiivé vývody z křemíkové destičky jsou zapojeny do Wheatstonova můstku.

Měřicí křemíková destička je často přiletována k nosné destičce ze skla a poté celý přípravek nalepen na kovovou destičku. Piezorezistivní snímače mají velmi dobré parametry, co se týče velikosti křemíkové membrány, stability čidla a malé hysterezi. Nevýhodou je citlivost na agresivní látky a vlhkost měřeného prostředí. Většina čidel je opatřena silikonovým gelem proti vlhkosti. Pro náročnější podmínky měření se čidlo chrání pouzdem a oddělovací membránou, která přenáší tlak pomocí nevodivé kapaliny na křemíkovou destičku. Nejběžnější konstrukce lze vidět na obr. 6.



Obr. 6 Řez snímače s oddělovací membránou[8]

Základní rozdělení tlakových senzorů podle principu měření:

Absolutní – okolní tlak (media) je vztažený k nulové hodnotě v komoře senzoru. Výstupní hodnota z tlakoměru odpovídá naměřené veličině.

Relativní – měří výchylku od požadované hodnoty nulového tlaku. Hodnota přivedeného tlaku na senzor odpovídá velikosti působené síly.

Diferenční – jde o rozdíl hodnot mezi dvěma hodnotami tlaku.

2.5 Mikrokontrolér PIC

Mikrokontrolér PIC je programovatelná polovodičová součástka vyráběná firmou Microchip, sestaven na harvardské architektuře (data nesdílí paměť s programem). Pro funkci procesoru je nutná taktovací frekvence, na které mikrokontrolér pracuje. Taktovací frekvence se vytváří pomocí RC členu nebo krystalu.

Mikrokontrolér obsahuje vstupně/výstupní porty a další možnosti periférií. Varianty periférií jsou např. A/D, D/A převodníky, čítače, vnitřní EEPROM paměti, rozhraní pro SPI nebo I²C komunikaci a další. Mikrokontrolér se programuje pomocí instrukční sady, která je pomocí programátoru nahrána do paměti mikrokontroléru. Jedná se o RISC sadu 35 strojových kódů jazyka assembler, kterým procesor rozumí. Procesor pracuje s binárními, decimálními nebo hexadecimálními čísly, která dokáže sečíst, odečíst nebo bitově posunout atd.

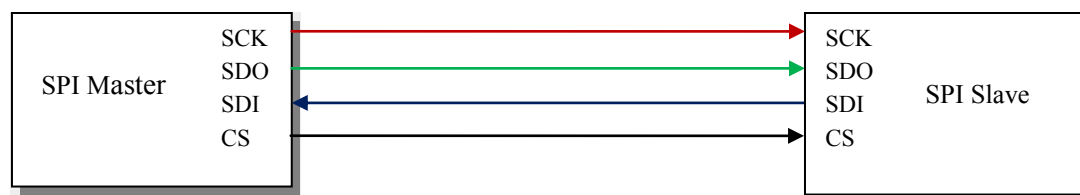
Dále se používá několik desítek registrů, které slouží pro nastavení periférií. Registry jsou místa v RAM paměti, která mají velikost 1Byte. Do těchto registrů můžeme zapisovat nebo z nich číst.

Firma Microchip nabízí několik řad PIC např. 8bitová řada – PIC10-18, 16bitová řada – PIC24, 32bitová – PIC32.

2.5.1 SPI

Externí sériová sběrnice SPI (Serial Peripheral Interface) je jedna z nejvíce používaných v digitálních zařízeních. Díky lehké naprogramovatelnosti a implementace do integrovaných obvodů má tato sběrnice velkou oblibu. SPI sběrnice je schopna komunikace mezi dvěma či větším počtem připojených zařízení. Přenosová rychlost je vysoká díky hodinovému signálu, který může dosáhnout až 70Mhz. Komunikace po SPI sběrnici je synchronní a umožňuje oboustranný přenos dat. Jeden uzel většinou vystupuje jako Master a druhý Slave. Master má funkci generátoru hodinového impulsu pro další uzly Slave. Výstup hodinového signálu se označuje SCK. Dvojice vodičů pro přenos dat se nazývají MOSI(Master Out, Slave In) a MISO(Master In, Slave Out). Posledním výstup je vodič SSEL (Slave select) pro výběr uzlu v režimu Slave.

Samořejmě se setkání s jiným označením pro tyto vodiče. V dokumentaci mikrokontroléru PIC se používají označení v režimu Master MOSI → SDI (Serial Data In), MISO → SDO(Seriál Data Out), SSEL → CS (Chip Select), u SCK zůstává název stejný. Výhodou této sběrnice je oboustranné zasílání dat, protože každý výstup má svůj datový vodič. Tím se nemusí řešit problém přepínání mezi vysíláním a příjmem dat.

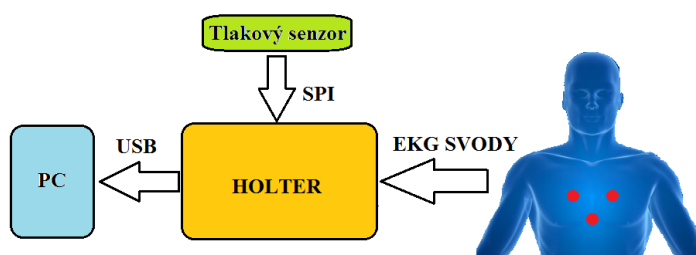


Obr. 7 SPI sběrnice v alternativním značení bez dalších periférií

3 Návrh zařízení

Práce vzniká ve spolupráci s PhDr. Mgr. Miloslavem Klugarem, Ph.D, který řeší problematiku reakční a adaptační změny aktivity autonomního nervového systému v souvislosti s hyperbarickým prostředím. Kapitola se zabývá výběrem součástek a optimalizací požadovaného zařízení pro měření biologických signálů ve freedivingu.

Pro měření srdeční aktivity a tlaku v hydrostatickém prostředí, bylo třeba opatřit pouzdro, které bude dostatečně tlaku odolné a malé pro uchycení na potápěči. S rozměrem pouzdra souvisel výběr tlakového senzoru s ohledem na umístění senzoru, rozsah a velikost požadovaného napájení.



Obr. 8 Návrh zařízení

3.1 Výběr holteru

Pro účel výběru zařízení byla využita bakalářská práce studenta Hanáka - Univerzální zařízení typu Holter s ukládáním dat na paměťovou kartu. Holter splňoval požadované parametry – digitální i analogové vstupy pro zakomponování senzoru, napájení dobíjecím akumulátorem, který má dostatečnou výdrž, pro uložení zaznamenaných dat, holter obsahuje modul s SD kartou. Data lze také přenést přes USB port. Spouštění holteru funguje skrze červené tlačítko.

3.2 Výběr tlakového senzoru

Na trhu existuje velké množství tlakových senzorů s různým účelem použití, které mají jak digitální tak i analogové výstupy. Existují také plně integrované senzory. V přístrojovém potápění jsou tyto senzory zabudované v „diving computers“. U těchto senzorů je potřebná dostatečně velká schopnost rozlišit změny tlaku a požaduje se velká míra integrovatelnosti.

Požadavky při výběru senzoru:

- Napájecí napětí: 2 – 3,6 V
- Vysoká rozlišovací schopnost

- Rozsah měření: až 30 bar
- Výstup: analogový nebo digitální
- Komunikace: SPI nebo jiná
- Velikost
- Cena

Vytvořené senzorové aplikace

Kromě modulů senzoru lze koupit „senzorové aplikace“ od firem HOPE RF a SensorOne. Tyto aplikace mají velký rozsah měření, a to až přes 25 bar s velkou rozlišovací schopností a přesností. Mají vytvořené obaly z nerez oceli.

K těmto aplikacím jsou dodávána přídatná zařízení pro funkci senzoru. Nevýhodou je vysoká pořizovací cena, velikost a nemožnost propojení s jinými zařízeními.

3.2.1 Porovnání tlakových senzorů

Do výběru tlakového senzoru bylo zahrnuto mnoho výrobců, kteří se zabývají tlakovými senzory v konečném výsledku, byly vybrány dva typy senzorů od výrobců HOPERF a Intersema. Na obrázku č. 9 je znázorněno jejich technické porovnání.

Parametry	HOPERF HSF2000-JK	Intersema MS5535-30C
Druh	Piezosensitive	Piezosensitive silicon
Napájecí napětí	5 - 10 V	2,2 - 3,6 V
Rozsah měření	0 - 20 bar	0 - 30 bar
Rozlišení	-	3,0 mbar
Výstup	Analogový	Digitální
Komunikace	Analogová	SPI
Teplotní kompenzace	NE	ANO

Obr. 9 Parametry senzorů

Technickými parametry, pro realizaci požadavku, nejvíce vyhovoval senzor firmy Intersema. Podrobnější parametry jsou uvedeny v tabulce č. 1.

Firma Intersema dodává tlakové senzory do potápěčských hodinek a počítačů. Senzor MS5535-30C má přídatnou aplikaci – a to měření vnější teploty. Tato aplikace je nezbytná pro teplotní kompenzaci senzoru, která umožňuje přesné měření.

Parametr	Znak	Hodnota
Maximální povolený měřený tlak při 25 °C	P	50 bar
Hodinový signál pro přenos dat	SCLK	max. 500 kHz
Hodinový signál pro vnitřní funkce	MCLK	30-35 kHz
Rozsah pracovní teploty	T	-40 – 125 °C
Maximální chyba v rozmezí napájení $V_{DD} = 2.2...3.6\text{ V}$	p=konst.	-40 +40 mbar
Rozlišení A/D převodníku	-	16 bit
Rychlost převodu	-	35 ms

Tab. 1 Bližší parametry senzoru firmy Intersema

3.3 Parametry PIC24FJ64GB002

Tento typ mikrokontroléru s 28 piny je zasazen do prototypu desky holteru. Programovacím nástrojem pro mikroprocesor je rozhraní MPLAB IDE, které obsahuje knihovny pro assembler, linker, kompilátor a debugger. Komunikace probíhá pomocí programátoru typu PICKIT 3 umožňuje programování a ladění.

- Pouzdro – 28 – DIP (7,5mm)
- Napájecí napětí – 2,0V až 3,6V
- Příkon – technologie XLP – technologie malého příkonu, odběr v režimu spánku v nano ampérech, při činnosti mikrokontroléru kolem 15μA.
- Architektura – 16bit, Modifikovaná Harvardská
- CPU – 16 MIPS
- Programovatelná paměť FLASH – 64KB
- Operační paměť RAM – 8KB
- Vnitřní oscilátor – 8 MHz, 4x PLL 32MHz
- Počet vstupů/výstupů – 19 I/O
- Čítače/časovače – 5x 16bit s programovatelným prescalerem
- Komparátory/PWM – 5x16bit výstup
- Převodníky – A/D 10bit – 9 kanálů
- Komunikace – USB 2.0 (Universal Serial Bus), 2 x SPI, 2 x I²C, 2 x UART [9]

3.4 Senzor MS5535-30C

3.4.1 Popis

Tlakový senzor MS5535-30C obsahuje přesné piezorezistivní tlakové čidlo a ADC- rozhraní IC (integrovaný čip). Tvoří je nemagnetický, nerezový kroužek O-ring, poskytující 16-ti bitová data tlaku a teploty závislé na napětí. Mimo to senzor obsahuje 6 koeficientů pro vysoce přesnou softwarovou kalibraci senzoru, které jsou uloženy ve vnitřní 64-ti bitové paměti PROM.

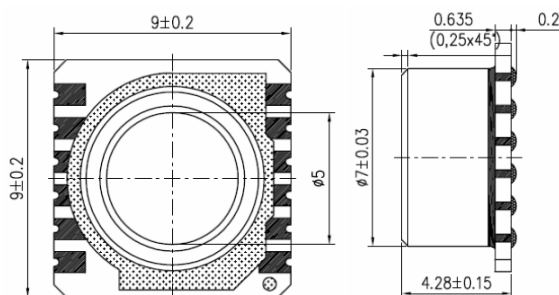
Senzor MS5535-30C má nízkou spotřebu napětí s automatickým vypínáním během nečinnosti. Komunikuje s mikrokontrolérem po SPI sběrnici(3-wire).

Využití

- Podvodní přenosné tlakoměry
- Potápěčské hodinky a počítače

3.4.2 Rozměry

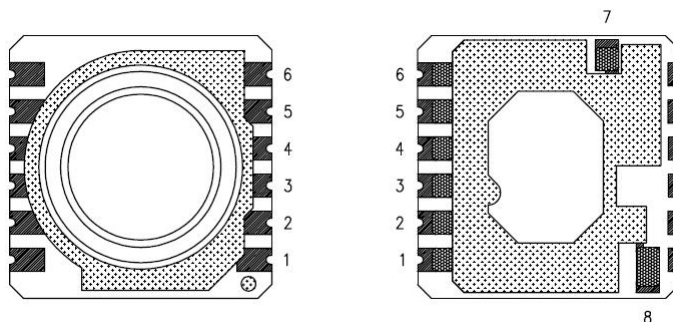
Předností senzoru je jeho velikost o rozměrech (šířka /výška /hloubka) 9 /9 /4,28 mm a průměr nerezového kroužku 5 mm. Rozměry velikost umožňuje pohodlné uložení do tlakového pouzdra. Nárys a bokorys senzoru s rozměry jsou zakresleny na obr. 10.



Obr. 10 Náčrty senzoru s rozměry[11]

3.4.3 Konfigurace pinů

Na obrázku č.11 je senzor s číselně označenými piny, které s pomocí tabulky č.2 informují o funkci pinů. Tyto údaje budou využity při programování komunikace mezi senzorem a mikroprocesorem. Vývody 7 a 8 nejsou používány.



Obr. 11 Konfigurace pinů MS5535-30C[11]

Název pinu	Pin	Typ	Funkce
GND	1	G	Zem
SCLK	2	Input	Hodinový signál pro přenos dat
DOUT	3	Output	Datový výstup
DIN	4	Input	Datový vstup
MCLK	5	Input	Hodinový signál pro vnitřní funkce(32.768kHz)
V_{DD}	6	Pozitive	Napájení
PEN	7	Input	Piny určeny pouze pro tovární kalibraci, nejsou zapojovány
PV	8	Negative	

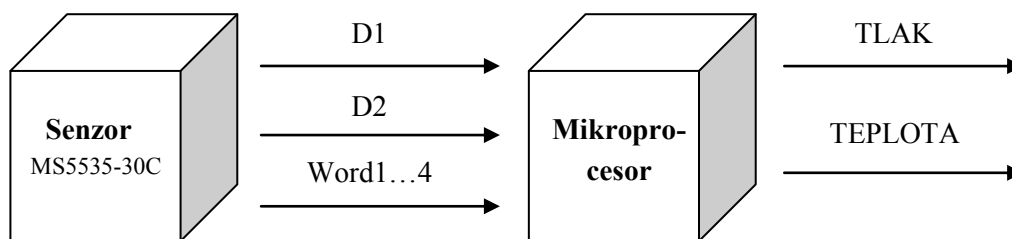
Tab. 2 Popis pinů senzoru MS5535-30C

3.4.4 Obecný popis funkce senzoru

MS5535-30C se skládá z piezo-rezistivního senzoru a IC rozhraní. Hlavní funkcí je přeměna nekompenzovaného analogového signálu z výstupu piezo-rezistivního senzoru do 16 bitové hodnoty.

- Naměřená 16 bitová hodnota tlaku „D1“
- Naměřená 16 bitová hodnota teploty „D2“

Výstupní napětí z tlakového senzoru je závislé na teplotě a je nutné kompenzovat tuto závislost. Kompenzace je prováděna v mikrokontroléru pomocí vytvořeného algoritmu firmou Intersema. Pro převod z analogové hodnoty tlaku a teploty do číslkové formy, je použit ADC převodník (sigma delta). Na obr. 12 lze vidět nutný postup pro získání kompenzovaných hodnot tlaku a teploty.



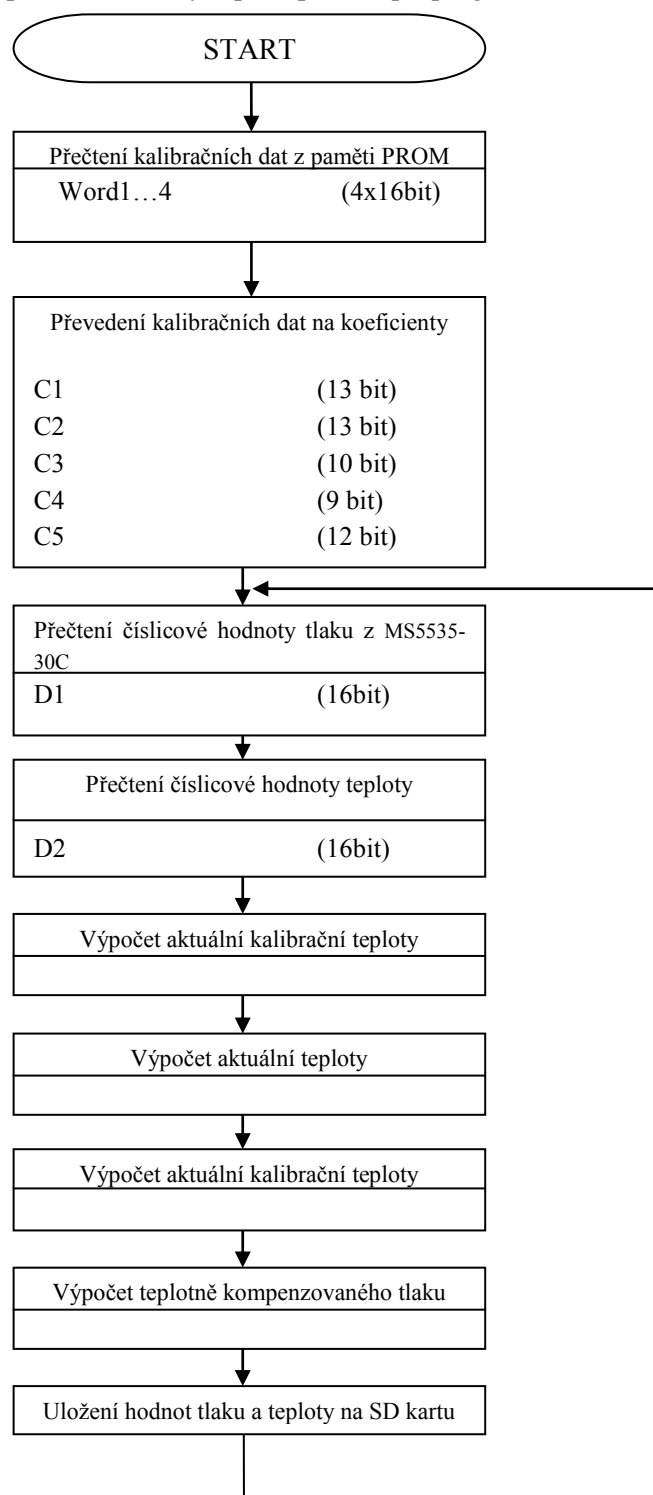
Obr. 12 Znáznornění komunikace mezi senzorem a mikroprocesorem

Tovární kalibrace

Každý modul je továrně kalibrován. Kalibrace je prezentována pomocí čtyř Wordů pro tlak a teplotu. Výsledek přepočtu Wordů je 6 koeficientů potřebných pro přesný výpočet hodnot tlaku a teploty. Tato kalibrační slova jsou uložena v 64-bitové PROM paměti. V následujícím obrázku vývojového diagramu lze vidět postup výpočtu hodnot tlaku a teploty.

3.4.5 Vývojový diagram

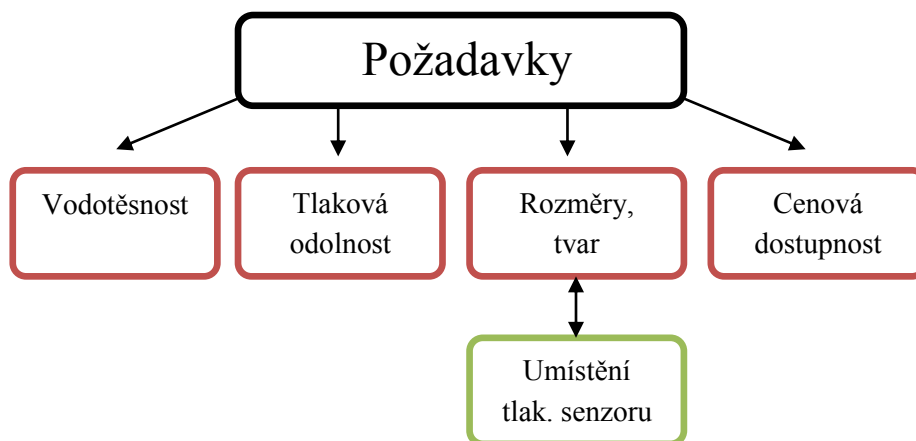
Na obrázku vývojového diagramu lze vidět proces výpočtu hodnot tlaku a teploty senzoru MS5535-30C, podle kterého bylo postupováno při programování.



Obr. 13 Vývojový diagram pro tlak a teplotu čtení a softwarové kompenzace

3.5 Výběr vodotěsného pouzdra pro holter

K realizaci speciálního holteru pro sledování biosignálů freedivera bylo nutné zajistit ochranu navrhované elektroniky. Musely být dodrženy následující požadavky viz. obrázek č. 14.



Obr. 14 Seznam požadavků pro výběr pouzdra

Vodotěsnost

Jeden z nejdůležitějších požadavků, zajištěn zajišť dostatečnou vodotěsnosti tak, aby nebyla poškozena elektronika, která se v ní nachází. Přesné dílenské zpracování, muselo zamezit vniku vody do pouzdra. Mnoho firem na českém trhu, se zabývá výrobou potápěčské techniky např. LOLA, IQsub. Specializují se na vodotěsná pouzdra pro velké hloubky. Většina obalů je vytvořena na míru pro určité aplikace např. svítilny, kamery, akumulátory aj.

Tlaková odolnost

Pouzdro muselo odolat tlaku okolo hodnoty 20 bar a více. Bylo důležité, aby pouzdro bylo předimenzováno na vyšší tlaky pro ochranu elektroniky.

Tlaku 20bar dosáhneme v hloubce 100m. Do hloubky 100m pod mořem se profesionální freediver dokáže běžně dostat.

Rozměry

Rozměry musely být přizpůsobeny freediverovi, tzn. neomezovat jej v pohybu, současně být dostatečně velké pro umístění digitálního senzoru, který bude zaznamenávat tlak při sestupu. Při připevnění na zápěstí či zátěžový opasek, muselo mít pouzdro vytvořeno speciální úchyty.

Cenová dostupnost

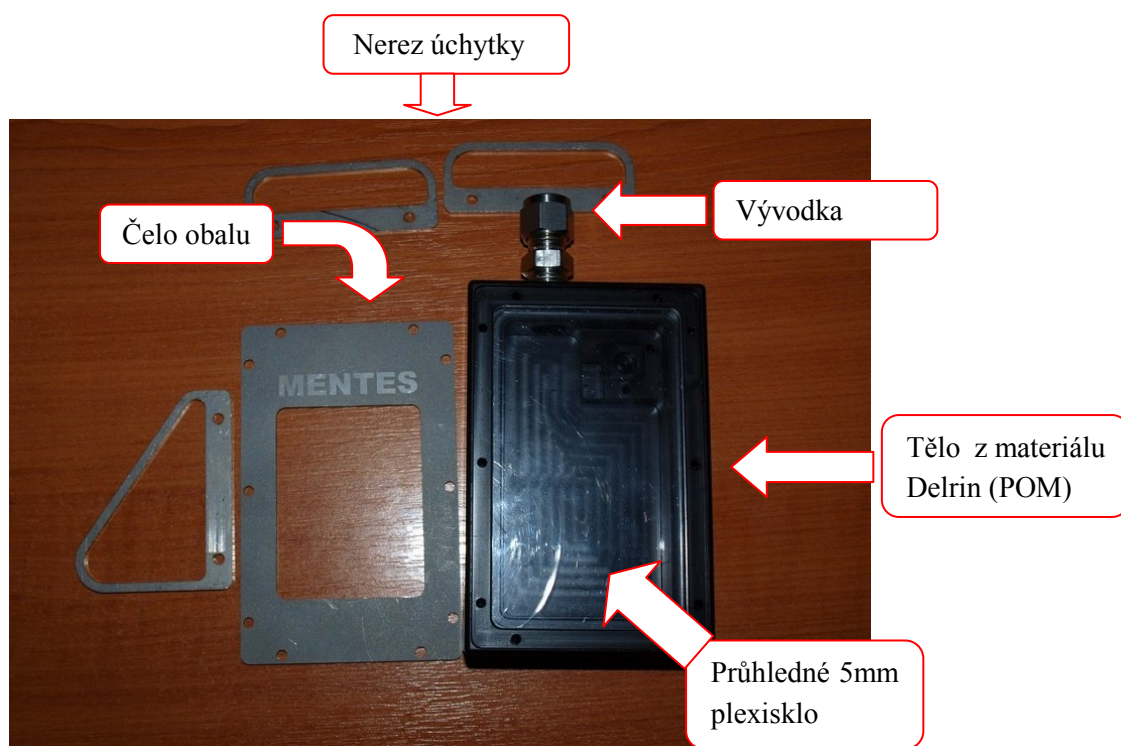
Ceny obalů pro potápěčskou techniku se pohybují v řádu tisícovek až desítitisícovek korun za pouzdro. Pouzdra vytvořená na míru stojí kolem desítek až stovek tisíc korun.

3.5.1 Obal MENTES

Dle požadovaných parametrů byl vybrán obal firmy MENTES, která se zabývá CCR rebreathery.

Pouzdro bylo vytvořeno na CNC fréze z materiálů Delrin® (POM), o rozměrech (110 x 75 x 25) mm. Jeho kovové části jsou vyrobeny z nerez oceli. Pouzdro testováno do hloubek až 200 m a má vývodku pro svody EKG.

Rozložené pouzdro a jeho části jsou ukázány na obr. 15 a na obrázku 16. Všechny díly se musí připevnit pomocí nerezových šroubků.



Obr. 15 Pouzdro holteru a jeho díly



Obr. 16 Místo pro usazení tlakového senzoru

- Čelo obalu – vyrobeno z nerez oceli, uprostřed čela je otvor pro display.
- Nerez úchytky – vyrobeny z nerez materiálu. Pomocí těchto úchytek si potápeč může obal připevnit na zápěstí nebo na pásek se zátěží.
- Vývodka – vyrobená firmou Swagelok z nerez oceli. Slouží pro vyvedení kabelu.
- Průhledné plexisklo – klasické plexisklo tloušťky 5 mm.

Delrin© (POM)

Registrovaná značka firmy DuPont. POM má velké využití v automobilovém průmyslu, domácích spotřebičích, sportovním vybavení, elektronice, spotřebním zboží a mnoha dalších aplikacích. Vyznačuje se nízkým koeficientem tření, vynikající kvalitou povrchu a zachováním mechanických parametrů i při velice nízkých teplotách, což ho předurčuje pro použití v mnoha průmyslových aplikacích.[10]

Vlastnosti:

- rozměrová stabilita
- vysoká pevnost, tuhost a rázová houževnatost
- vysoká odolnost vůči únavovému lomu
- dobré elektroizolační schopnosti
- použitelnost i při velmi nízkých teplotách

4 Technická realizace

V této části je popsáno propojení mikroprocesoru a tlakového senzoru a vytvoření tištěného spoje pro upevnění digitálního senzoru k obalu.

4.1 Propojení mikroprocesoru s tlakovým senzorem

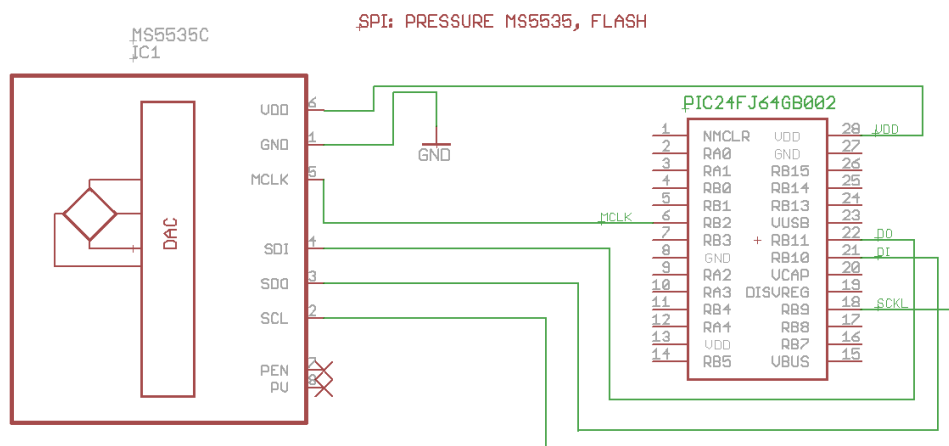
Prostudováním schématu zapojení holteru z bakalářské práce - Univerzální zařízení typu Holter s ukládáním dat na paměťovou kartu, a přečtením konfigurace pinů tlakového senzoru viz. předchozí tabulka č. 3, bylo možné propojit senzor a mikroprocesor.

Označení a význam použitých pinů mikroprocesoru je vysvětlen v tabulce č. 4.

Piny	Označení	Název	
22	DO	Data Out	Datový výstup
21	DI	Data In	Datový vstup
18	SCKL	Seriál Clock	Hodinový signál SPI komunikace
6	MCKL	Master Clock	Hodinový signál pro funkci senzoru
28	VDD	Voltage pozitivne	Napájení

Tab. 3 Označení pinů mikroprocesoru

Obrázek č. 17 znázorňuje schéma propojení PIC24FJ64GB002 a senzoru MS5535C-30C

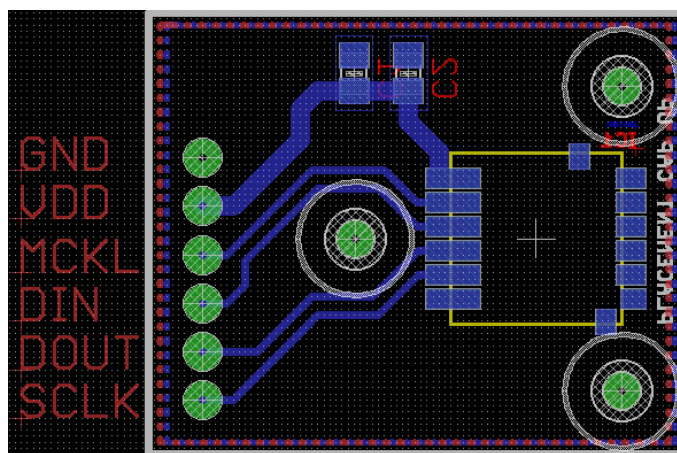


Obr. 17 Schéma propojení senzoru a mikroprocesoru

4.2 Návrh a výroba plošného spoje pro tlakový senzor

Byl vytvořen plošný spoj, na kterém byla připevněna destička tlakového senzoru. Tento krok byl nezbytný pro oporu a připevnění senzoru v pouzdru. Byl využit existující návrh plošného spoje, který dodala firma Mentec. Pomocí softwaru Eagle byla navržena destička plošného spoje. Pro náš záměr musel být plošný spoj připravován. Velikost, rozteč a rozměry děr pro uchycení tištěného spoje zůstaly zachovány. Propojení bylo uskutečněno plochým více žilovým, páskovým vodičem. Vytvořením desky plošného spoje bylo usnadněno připájení propojovacího

vodiče. Vyvedením pinu na plošném spoji se usnadnilo připájení komunikačních vodičů. Na obr. 18 lze vidět způsob umístění senzoru na desku.



Obr. 18 Deska plošného spoje

Na vodivou cestu vývodu pinu V_{DD} byly přidány paralelně dvě kapacity C1, C2 o celkové velikosti $10,1\mu F$. Firemní dokumentací senzoru bylo doporučeno přidat dvojici tantalových kapacit pro zajištění stabilizace napětí během převodu dat. Tímto lze dosáhnout co největší přesnosti měření. Destička má rozměry 22 mm na výšku a 78 mm na šířku. Otvory pro uchycení mají průměr 3,1 mm. Během letování bylo nutné dodržet teplotní profil, který zajišťuje ochranu senzoru proti zničení.

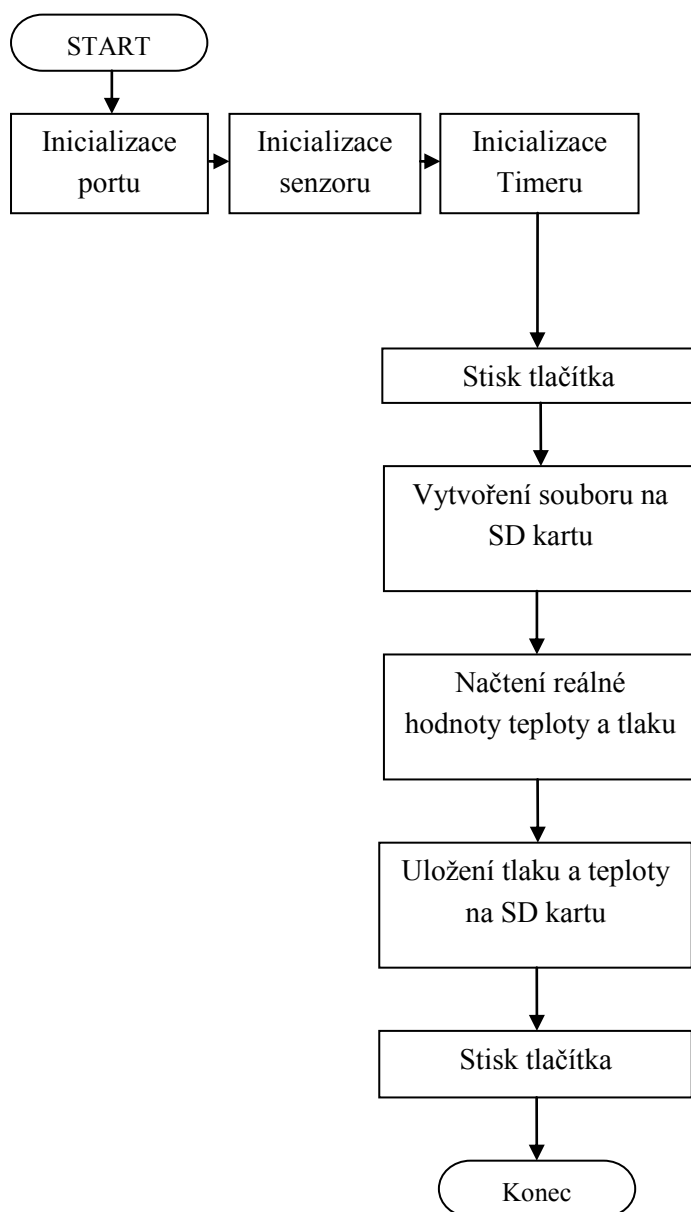
Deska plošného spoje byla vyrobena firmou 7Marsyas.

5 Softwarová realizace

Softwarová část se zabývá vytvořením komunikačního protokolu mezi senzorem a mikroprocesorem. Vyčítá, zpracovává data ze senzoru a následně ukládá data pomocí mikrokontroléru na paměťové medium.

5.1 Obecný diagram měření tlaku a teploty

Na obrázku č. 19 obecného diagramu je znázorněn jeden cyklus měření od začátku připojení napětí až po uložení dat a konec. Přesný popis úkonů, které je třeba vykonat pro měření hodnot, bude popsán v dalších podkapitolách.



Obr. 19 Obecný diagram funkce programu měření tlaku a teploty

5.2 Komunikace mikrokontroléru a tlakového senzoru

V první fázi byla vytvořena SPI komunikace pomocí standartního protokolu, který nabízí řada PIC24Fxxx. Bylo využíváno registrů, které nastavují RP-piny (překonfigurovatelné piny) mikroprocesoru. Tyto překonfigurovatelné piny lze naprogramovat a použít pro vstup, výstup nebo hodinový impuls SPI sběrnice. Díky specifickým požadavkům komunikace po SPI senzoru, bylo nutné vytvořit tuto komunikaci softwarově – znamená ovládat každý bit jednotlivě. Pro tyto účely byla využita dokumentace firmy Intersema.

Na začátku vytváření komunikace senzoru s mikroprocesorem bylo zapotřebí nastudovat problematiku mikroprocesorů a také příloženou dokumentaci k mikroprocesoru PIC24Fxxx. Kód byl vytvářen pomocí jazyka C v uživatelském prostředí MPLAB X. Soubor Sensor.c obsahuje funkce pro řízení komunikace mezi procesorem a senzorem. K tomuto souboru byly vytvořeny hlavičkové soubory include.h a Sensor.h, které obsahují definice pinů, různá nastavení, makra a prototypy funkcí pro kompilátor. Bez hlavičkových souborů by procesor nepřečetl funkce, které jsou napsány v .c souborů.

Nastavení mikroprocesoru

Vybrané piny na mikroprocesoru, byly určené pro SPI komunikaci, byly nastavovány pomocí registru TRISB.

Nastavení I/O portu pro sběrnici SPI bylo vytvářeno pomocí funkce SPI2_INIC (RB11 = výstup, RB9 = výstup, RB10 = vstup).

5.3 Komunikace senzoru po SPI

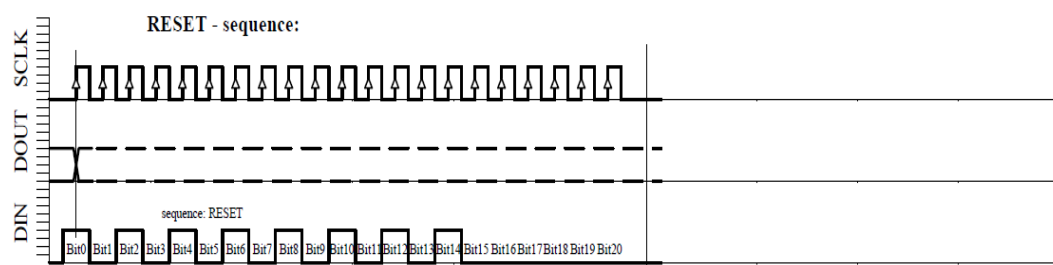
5.3.1 Vytvoření hodinového impulsu MCKL

Nutnou součástí pro funkci senzoru, bylo přivedení Master Clock o frekvenci 32 kHz, která se používá pro vnitřní výpočty. Tato frekvence byla vytvořena pomocí 16 bitového Timeru a Interruptu (Přerušení). Ve funkci Inicializace_T1 jsou obsažena nutná nastavení Timeru a Interruptu. Poté byla vytvořena obsluha Interruptu, do které vchází mikroprocesor při každém přetečení registru RP1. Registr RP1 je nastaven na hodnotu 0xFA. Toto číslo nám určuje periodu přetečení 30,51 μ s. Po přetečení se nastaví interrupt flag, který informuje, že nastalo přerušení. Toto přerušení obsahuje makro MC_TOGGLE, které mění hodnotu na výstupu portu MCKL a vytváří tak hodinový impuls o frekvenci 32 kHz. Tělo interruptu musí obsahovat mazání interrupt flagu. Pokud by bit nebyl smazán, procesor zůstane v tomto přerušení.

5.3.2 Získání kalibračních hodnot

Po nastavení pinů byla vytvořena inicializace senzoru funkcí `Inicializace_senzoru` obsahující sekvenci potřebnou pro správné měření senzoru. Nejprve bylo potřeba provedení resetu senzoru a poté zaslat sekvenci bitů pro vyčtení kalibračních hodnot (Word).

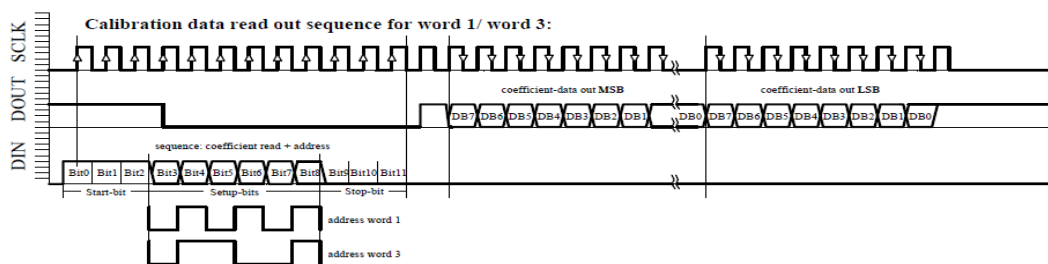
```
reset_senzoru()
```



Obr. 19 Sekvence resetu [11]

Reset sekvence byl potřebný před vyčtením kalibračních hodnot na počátku měření. Sekvence obsahuje 21 bitů, jak lze vidět na obr. 19. Tento reset má za účel vytvořit novou komunikaci mezi senzorem a mikroprocesorem pokud byla přerušena. Také bylo doporučeno vložit reset sekvence mezi každou konverzi (převodem), aby se zabránilo přerušení komunikace.

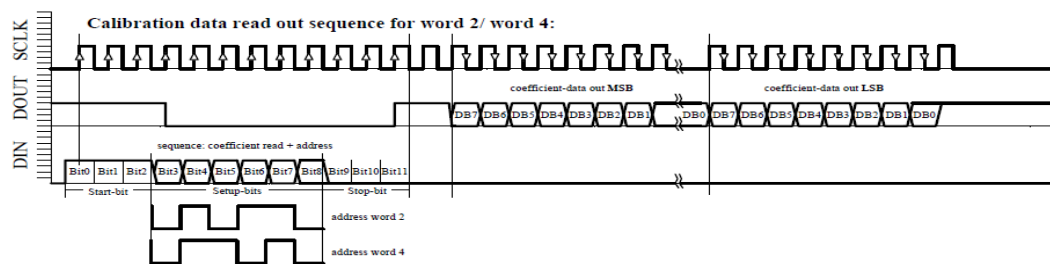
```
word1 = ziskej_konfiguracni_word(1)
word3 = ziskej_konfiguracni_word(3)
```



Obr. 20 Sekvence pro získání kalibračních hodnot 1/3 [11]

Na obrázcích je znázorněna sekvence pro vyčtení kalibračních Wordů, kde zobrazený port senzoru DIN na obr. 20 znázorňuje 12 přijatých bitů do senzoru po náběžné hraně SCLK od mikroprocesoru. Poté mikroprocesor čeká jeden hodinový impuls, než začne senzor vysílat data z výstupu DOUT na sestupné hraně hodinového impulsu. Příjem hodnot senzoru obsluhuje funkce `vyciti_16_bitu_ze_senzoru`. Wordy o velikosti 32 bitů a jsou ukládány do globálních proměnných `long word1, word2, word3, word4`.

```
word2 = ziskej_konfiguracni_word(2)
word4 = ziskej_konfiguracni_word(4)
```

Obr. 21 Sekvence pro přečtení kalibračních hodnot 2/4 [11]

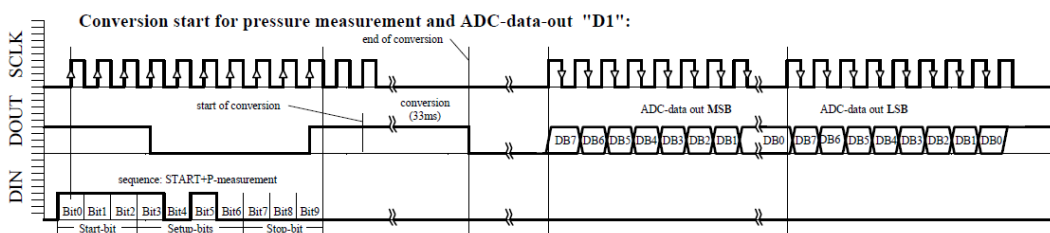
Obrázek č. 21 zobrazuje sekvenci pro vyčtení kalibračních hodnot 2 a 4. Po přijetí dat zasílá jeden hodinový impuls pro ukončení komunikace.

5.3.3 Převod kalibračních hodnot

Po získání tovární kalibrace je bylo potřeba tyto hodnoty přepočítat do 32bitových koeficientů C1 – C6. Tyto koeficienty obsahují např. offset tlaku a teploty. Získané Wordy byly převedeny do 6 koeficientů pomocí funkce `Prevod_Wordu` a hodnoty ukládány do globální proměnné typu pole o velikosti C[6] rozsahu 32bitů.

5.3.4 Získání hodnot tlaku a teploty

Pro získání teploty a tlaku slouží funkce `dostan_D1`, `dostan_D2` datového typu long. Kde D1 označuje hodnotu tlaku a D2 teplotu. Sekvence pro získání tlaku z senzoru je vyobrazena na obr. 22.



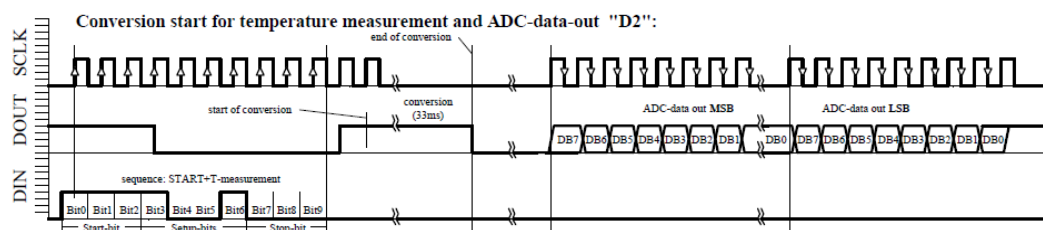
Obr. 22 Získání hodnoty D1[11]

Pro získání hodnot tlaku musí být poslána sekvence 10-ti bitů v hodnotě 0x3D0 po nastupné hraně hodinového impulsu dokud na vytupu DI mikroprocesoru nebude detekována jednička. Pokud ne, zůstává v cyklu while.

```
while(PORTBbits.RB10 == 0) - hodnota DOUT ze senzoru
```

Jestliže byla detekována jednička, vyšlou se dva hodinové impulsy a poté proběhne vnitřní převod hodnot v senzoru. Během vnitřního převodu dat senzoru, hodiny SCLK zůstanou na nulové hladině. Detekci nuly na vstupu mikroprocesoru bylo ukončen převod. Poté senzor začne vysílat data po sestupné hraně hodin SCLK.

Sekvence pro získání teploty na obr. 23 pomocí funkce `dostan_D2` je velmi podobná jako u předešlého případu.



Obr. 23 Získání hodnoty D2[11]

Obrázek č. 23 znázorňuje sekvenci pro vyčtení hodnoty teploty. Na rozdíl od minulé sekvence se zde posílá 10bitů o hodnotě 0x3C8. Poté mikroprocesor čeká na výstupu hodnotu 1 ze senzoru. Další postup je stejný jako v předešlém vyčtení tlaku.

5.3.5 Výsledný výpočet tlaku a teploty

Sekvence pro příjem hodnot tlaku, teploty a pro převod wordů byla zakomponována do společné funkce `nacti_a_prepocitej_xxx`. Pro ukázkou byla použita funkce `nacti_a_prepocitej_tlak`, která je totožná s funkcí `nacti_a_prepocitej_teplo`. Výpočet reálných hodnot tlaku a teploty pro řadu senzoru MS5535, který byl použit z knihoven typu AN502 byl chybný. Proto bylo nutné nalést jiný typ knihoven, které správně přepočítají hodnoty. Po nalezení vhodných knihoven byly otestovány pomocí referenčních hodnot, které se nacházely v datasheetu senzoru. Během testování byla objevena další chyba v řádku výpočtu konečné hodnoty tlaku.

```
p = sens * (d1-off)/4069+1000;
```

Tento přepočet byl nalezen v nové knihovně pro řadu senzoru MS5535. Podle datasheetu senzoru na místě hodnoty 4069 se má nacházet číslo $2^{11} = 2048$. Po opravě chybné konstanty celkový přepočet byl správný.

Funkce `vypocet_tlaku` obsahuje přepočty hodnot koeficientů a jiných parametrů. Po výpočtu, funkce navrácí 32 bitovou hodnotu tlaku v mbar. Viz následující kód.

```
long vypocet_tlaku(long d1,long d2)
{
    long ut20,off,sens,p,dut;
```

```

    ut20 = 8*C[4]+10000;
    dut = d2-ut20;
    off = C[1]+(C[3]-250)*dut/4096+10000;
    sens = C[0]/2+(C[2]+200)*dut/8192+3000;
    p = sens * (d1-off)/2048+1000;
    return(p);
}

```

Jako v předešlém případě, nalezená dokumentace typu ANS502 pro přepočet teploty, byla špatná. Po nalezení odpovídající dokumentace byl použit relativně správný přepočet, který obsahoval chybný algoritmus výpočtu.

```

if (dut<0) dut2 = dut - (dut/128*dut/128)/4;
else dut2 = dut - (dut/128*dut/128)/8;

```

Po odstranění chybných řádků kódu funkce pro výpočet teploty fungovala korektně. Finální funkce výpočtu vypadá takto:

```

long vypocet_teploty(long d2)
{
    long temp,dut,ut20;

    ut20 = 8*C[4]+10000;
    dut = d2-ut20;
    temp = 200+dut*(C[5]+100)/2048;
    return(temp);
}

```

Návratová hodnota odpovídá hodnotě desetkrát větší než je skutečná.

Proměnné C[n] jsou koeficienty vypočtené pomocí funkce Prevod_Wordu, která byla popsána v předešlé části. Jde o 32-ti bitovou proměnnou typu pole.

5.3.6 Uložení hodnot tlaku a teploty na SD kartu

Měření tlaku začíná po stisku tlačítka na holteru. Nejdříve se však vytvoří globální proměnné pro uložení hodnoty tlaku a teploty s deklarací:

```

long real_tlak, real_teploata = 0;

```

Tyto proměnné se poté ukládají do 16-bitového pole MojeData[8]. Bylo zapotřebí vytvořit soubor, do kterého se bude ukládat pole mojeData.

```

ptSoubor = FSfopen(SOUBOR,WRITE);

```

Příkaz FSfopen vytváří a povoluje zapisování do proměnné s názvem SOUBOR. Proměnná SOUBOR byla definována jako binární soubor s názvem senzor. Binární soubor byl zvolen pro svojí paměťovou nenáročnost.

Pro nastavení periody, jak často má mikroprocesor získat hodnotu tlaku, slouží funkce, která se skládá z podmínky `if (nactene_pulzy > 32000)`.

Instrukce využívá vytvořeného interruptu, který slouží jako řídicí hodinový impuls (MCKL) pro senzor. Každým vstupem do interruptu se inkrementuje globální proměnná `pocitani_pulsu`, kterou pomocí funkce `get_pulsy` voláme. Příkazem `if` a jeho podmínkou nastavujeme, jak často vstupujeme do bloku příkazů, který obsahuje následující funkce:

```
if (nactene_pulzy > 32000)
{
    reset_senzoru();
    real_tlak = nacti_a_prepocitej_tlak();
    real_teploata = nacti_a_prepocitej_teploatu();

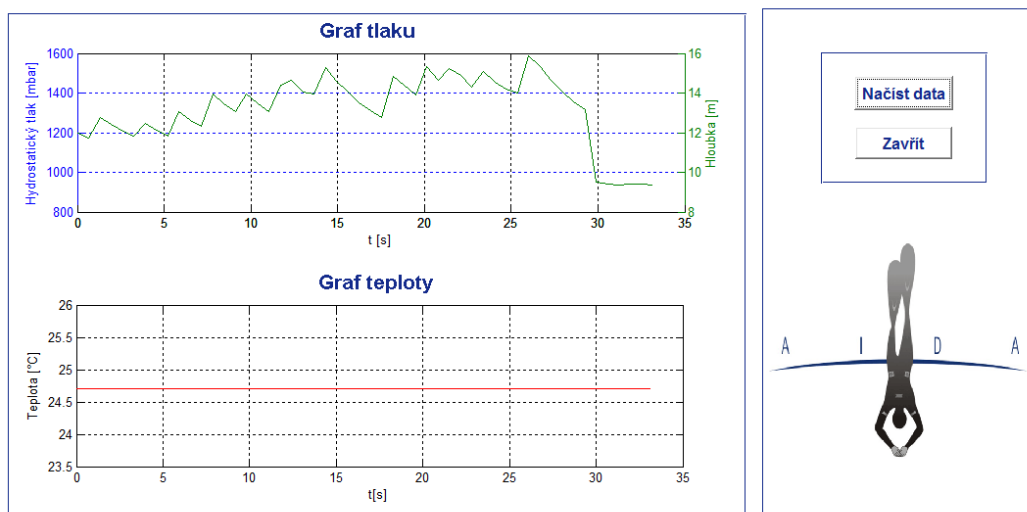
    mojeData[0] = real_tlak;
    mojeData[1] = real_teploata;
    mojeData[2]=0;
    mojeData[3]=0;
    mojeData[4]=0;
    mojeData[5]=0;
    mojeData[6]=0;
    mojeData[7]=0;
    FSfwrite((void*) mojeData, 16, 1, ptSoubor);
    vynulovani_pulsu();
}
```

Hodnota 32 000 umožňuje vnoření do těla podmínky každých 640ms.

Blok obsahuje algoritmy pro získání reálných hodnot tlaku a teploty. Do vytvořeného pole `mojeData[n]` vkládáme na definované pozice hodnoty ze senzoru. Bylo určeno, že na nultou pozici bude uložena hodnota tlaku a na první pozici hodnota teploty. Ostatní pozice jsou zaplněny nulami pro přehlednost dat a pro budoucí snadnější zpracování hodnot. Příkaz `FSfwrite` ukládá `mojeData` do určeného souboru, který byl vysvětlen výše.

6 Zpracování naměřených dat

Potřeba graficky znázornit naměřená data v závislosti v čase, vyžadovala vytvoření uživatelského rozhraní, kde si uživatel může zvolit soubor k vykreslení. Pro zpracování naměřených hodnot bylo využito matematicko-simulačního programu Matlab. Software Matlab nabízí několik nástrojů např. Simulink, GUIDE, Toolbox. Byl vybrán nástroj GUIDE pro grafické rozhraní.



Obr. 24 Uživatelské rozhraní

Na obr. 24 v pravém horním rohu jsou tlačítka pro načtení dat a ukončení programu. V levé části programu jsou umístěny osy pro vykreslení načtených dat. V grafu tlaku vynášíme na osu x čas a na osu y vynášíme hodnoty dvě, a to na levé straně hydrostatický tlak p [mbar], na straně pravé přepočten tlaku na hloubku h [m]. Přepočten byl vytvořen pro orientaci uživatele, v jaké hloubce se potápeč během měření nacházel. Pod grafem tlaku jsou osy pro vynesení závislosti teploty na čase. Teplota je udávána ve stupních Celsia. Měření teploty nebylo naším primárním cílem, ale informace o měnící se teplotě během ponoru může být přínosná.

6.1 Software pro nahrání a vykreslení naměřených dat

Načítání dat bylo prováděno pomocí tlačítka Načíst data. Po stisku se zobrazí okno pro výběr dat.

```
[FileName, PathName] = uigetfile('*.bin','Volba souboru')
if(FileName)~=0
    data = fullfile([PathName, FileName]);
    fid = fopen(data);
    hodnoty = fread(fid,'uint16','l');
else
    warndlg('Nulová hodnota dat','Chyba')
end
```

Nahraná data byla uložena do proměnné s názvem `hodnoty`. Soubor `hodnoty` je vektor hodnot.

```
tlak=hodnoty(1:8:end)
teplota=hodnoty(2:8:end)
```

Tímto kódem se vybírají ze souboru hodnoty tlaku a teploty. Ve vektoru `hodnot` se naměřený tlak nachází od první pozice na každé osmé. U teploty od druhé pozice každá osmá hodnota. Toto pořadí vychází z volby uložení hodnot tlaku a teploty do pole `MojeData[n]` viz. kapitola **Komunikace senzoru po SPI**.

```
for i=1:1:(size(tlak))
    pocitadlo(i,1)=k;
    k=k+0.65;
end
```

Z důvodu neoznačení naměřených hodnot v souboru pomocí času nebo číselné hodnoty. Nebylo možno určit, kdy byl vzorek zaznamenán. Pomocí výše zmíněného algoritmu byl převeden počet vzorků na skutečný čas zaznamenání hodnoty.

Příkazem `plotyy` se vytvořily dvě osy y. Pokud načtená data budou nulové hodnoty, podmínka `if` obslouží chybu a ukáže chybové okno.

```
if (tlak)~=0
    %blok příkazů
else
    warndlg('Nulová hodnota dat','Chyba')
```

7 Testování

V této části bylo nutné otestovat tlakový senzor MS5535C-30C. Pomocí měřicího řetězce získat několik hodnot a porovnat s referenčním tlakem. Kalibrace byla prováděna pomocí softwaru uvedeného v kap. 5.3.2.

Použité vybavení

- simulátor tlaku RIGEL MEDICAL Uni-SiM
- snímač tlaku MC
- tlaková manžeta
- propojovací hadičky
- holter
- napájení holteru – PICKit 3
- SD karta 512 MB
- stolní PC
- láhev s vodou

7.1 Zapojení měřicího řetězce

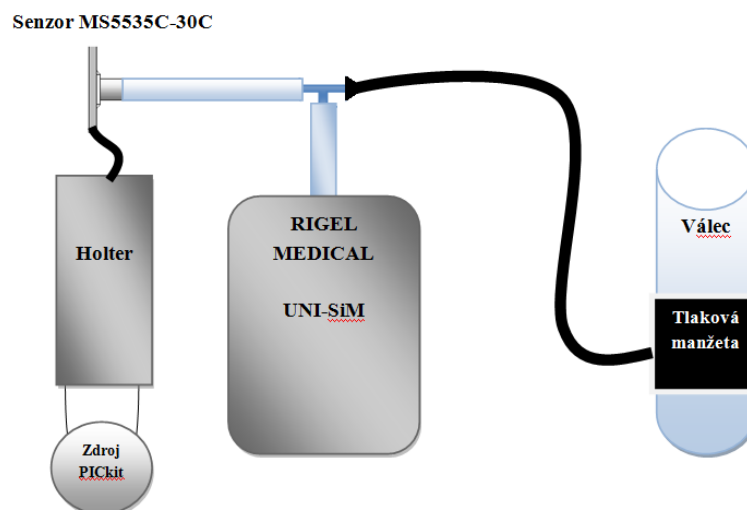
Měřicí řetězec se skládá ze zdroje napětí, senzoru, simulátoru tlaku a hadiček propojených s manžetou. Manžeta byla připevněna k pevnému válci (láhvi s vodou), která se pod tlakem bude deformovat. Konec manžetové hadičky byl připojen k roztrójce, která jednou stranou byla připojena k hadičce s připevněným senzorem a druhou stranu s k simulátoru tlaku. Snímač tlaku byl napájen z programovacího zařízení PICKit.

Průběh měření

Po zapojení měřicího řetězce byl na tlakovém simulátoru nastaven program *Static Pressure Test*. Po spuštění tohoto testu přístroj začal tlakovat měřicí řetězec na námi požadovanou hodnotu 150 mmHg (199,983 mbar). Po dosažení požadované hodnoty tlaku ukončil vytváření většího přetlaku. Poté byl přístroj v klidu a vykresloval na displayi průběh tlakové křivky.

Měření probíhalo takto:

- 1) Stisk tlačítka holteru – nahrávání hodnot tlaku.
- 2) Odpočet 3 sekund – doba od začátku měření tlaku do počátku spuštění činnosti programu STP.
- 3) Spuštění programu STP
- 4) Odpočet 10 sekund – po dosažení požadovaného tlaku odpočítáno 10 sekund.
- 5) Ukončení testu – po uběhnutí 10 sekund ukončení programu a automatické upuštění přetlaku v manžetě.

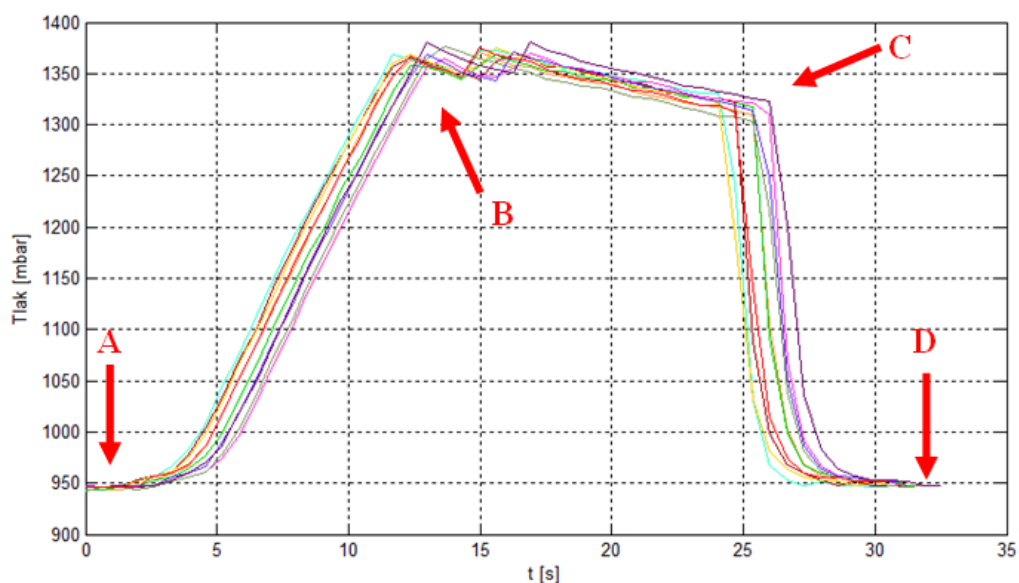


Obr. 24 Náskres měřicího řetězce

7.2 Výsledky měření

Všech deset měření tlakovým senzorem bylo zaznamenáno na SD kartu. Nahraná data byla vykreslena pomocí softwaru Matlab.

Obr. 26 znázorňuje grafickou závislost měřeného tlaku na čase. Bylo provedeno celkem 10 měření, pro které byl stanoven stejný postup. Do tabulky č. 4 byly zaznamenány hodnoty tlaku senzoru na počátku měření (A), při dosažení hodnoty 150 mmHg (referenční hodnota) tlakovým simulátorem (B), v době 10 sekund po dosažení referenční hodnoty (C) a na konci měření (D).



obr. 26 Graf závislosti tlaku na čase

	A	B	C	D
	[mmBar]	[mmBar]	[mmBar]	[mmBar]
1. měření	944	1363	1310	953
2. měření	948	1368	1331	946
3. měření	944	1358	1317	948
4. měření	946	1366	1321	948
5. měření	946	1364	1310	946
6. měření	942	1377	1303	948
7. měření	946	1360	1321	948
8. měření	948	1368	1314	951
9. měření	946	1380	1323	948
10. měření	946	1367	1312	951

tab. 4 Tabulka naměřených hodnot

Tímto měřením měla být stanovena přesnost měření senzoru. Z důvodu špatné těsnosti hadiček, nemohla být zjištěna přesnost měření tlakového senzoru, vůči referenčnímu přístroji.

8 Závěr

Cílem této práce bylo vytvořit měřicí zařízení pro monitorování biologických signálů freedivera. Nejdříve bylo nezbytné nastudovat si problematiku měření biosignálů v hydrostatickém prostředí. Po uvážení možností měření různých biosignálů, bylo vybráno snímání EKG s průběžným záznamem tlaku pod vodou. Jako měřicí EKG zařízení byl použit holter vytvořený v bakalářské práci studenta Hanáka. Při měření tlaku holter slouží jako sběrné zařízení, pomocí kterého se uloží naměřená data na SD kartu. Pro zajištění ochrany elektroniky jsem vybral vodotěsné a tlaku odolné pouzdro. Výběr obalu jsem podmínil cenovou dostupností, náročností vnějšího prostředí a jeho uložením na těle freedivera. Těmto požadavkům vyhovovala firma Mentos, která se zabývá měření plynů v CCR rebreatherech.

Firmou Mentos byl doporučen výrobce senzorů tlaku s názvem Intersema. Z nabídky senzorů firmy Intersema, jsem vybral typ MS5535-30C. Jde o digitální senzor, který měří absolutní tlak a teplotu. Pro komunikaci senzor využívá SPI sběrnici. Propojení mezi senzorem a holterem jsem vytvořil přes SPI rozhraní. Pro zajištění senzoru v pouzdru jsem vytvořil destičku plošného spoje, na které je přilepen senzor. Destička plošného spoje umožnila vyvedení pinů a snadnější přilepování propojovacího vodiče pro komunikaci s holterem.

Po vyřešení technické stránky jsem se dále věnoval programování komunikace a uložení naměřených hodnot tlaku a teploty. Funkce pro přepočty tlaku a teploty bylo nutné opravit, protože byly chybně vytvořeny v datasheetu senzoru. V poslední části bakalářské práce jsem vytvořil uživatelské rozhraní, ve kterém lze nahraná data zobrazit do grafu. Funkcí senzoru jsem otestoval pro 10 měření na měřicím řetězci. Testováním byla dokázána přesnost měření tlakového senzoru typu MS5535-30C, takže tento senzor je vhodný pro použití kontinuálního snímání hydrostatického tlaku během ponoru freedivera.

Moje práce je připravena pro pokračování ve vytváření měřicího řetězce pro kontinuální měření tlaku a EKG. Pro komerční využití je třeba do budoucna vylepšit stávající formu holteru např. bezdrátové přenosy dat.

9 Literatura

- [1] martin Štěpánek [Online]. ©2009,[cit. 16-1-2012]< <http://www.martinstepanek.com/cgi-bin/album.pl?photo=Photos/Freediving/F0005.jpg/>>
- [2] AIDA CZECH REPUBLIC [Online]. ©2007-2010,[cit. 16-1-2012] <<http://www.aida-czech.cz/index.php?stranka=discipliny>>
- [3] HORÁČEK, Jaroslav. Anatomie pro bakalářské studium se zdravotnickým zaměřením. Ostrava: VŠB skriptum, 2006. 113 s.
- [4] IMRAMOVSKÝ, Martin. Zdravotnické elektronické přístroje I. Ostrava: VŠB skriptum, 2007. 330 s.
- [5] JIRÁK, Zdeněk; Kolektiv. Fyziologie pro bakalářské studium na FZS OU. Ostrava: OSU skriptum, 2009. 250 s.
- [6] EKG Holter [online]. [cit. 16-1-2012]. < http://www.kardioprachatice.cz/pages/ekg_holter/>
- [7] BEJČEK L., VACULÍK J. Snímače tlaku – principy, vlastnosti, a použití (část 1). [online]. [citováno 29-4-2012] < http://www.odbornecasopisy.cz/index.php?id_document=42719>
- [8] BEJČEK L., VACULÍK J. Snímače tlaku. [online]. [citováno 29-4-2012] < http://web.vscht.cz/kadleck/aktual/MT_INaRP/Tlak2_au070725.pdf>
- [9] HANÁK V. Univerzální zařízení typu Holter s ukládáním dat na paměťovou kartu. Ostrava: VŠB, Technická univerzita Ostrava. Fakulta elektrotechniky a informatiky, 2011 45 s. Vedoucí bakalářské práce Ing. Martin Stankuš.
- [10] Dupont [online]. [cit. 16-2-2012]. <http://www2.dupont.com/Czech_Republic_Country_Site/cs_CZ/Products_and_Services/Products/delrin.html>
- [11] MS5535-30C 30 bar Pressure Sensor Module [online]. 2011. 21s. [cit. 29-4-2012] <<http://www.meas-spec.com/downloads/MS5535-30C.pdf>>

10 Přílohy

Příloha I.	Kód souboru Senzor.c
Příloha II.	Kód souboru Sensor.h
Příloha III.	Kód souboru include.h
Příloha IV.	Kód souboru pic24f_demo.c
Příloha V.	Kód Matlab uživatelské rozhraní

Příloha I.

```
/*Funkce senzoru MS5535-30C pro komunikaci a zpracování hodnot tlaku*/
#include "Compiler.h"
#include "GenericTypeDefs.h"
#include "HardwareProfile.h"
#include "include.h"
#include "Senzor.h"
long word1, word2, word3, word4 = 0; //wordy pro vypocet
long C[6];
UINT32 pocitani_pulsu;
/*Time1 interrupt pro pin Master clock senzoru o frekvenci 32kHz */
void __attribute__((interrupt,no_auto_psv)) _T1Interrupt(void){
    MC_TOGGLE();
    T1_RESET_INTR_FLAG;
    pocitani_pulsu++;
}
void Inicializace_T1(void){
    // nastaveni timer
    MC_INIC();
    REGISTR_PR1();
    T1_VNITR_OSC;
    T1_PREDDEL_1;
    T1_RESET_INTR_FLAG;
    T1_INTR_ZAP;
    T1_ZAP;
}
UINT32 get_pulsy(void){
    return(pocitani_pulsu);
}
void vynulovani_pulsu(void)
{
    pocitani_pulsu=0;
}
/* ----- */
/* ----- Vypocet tlaku a teploty ----- */
/* ----- */
long vypocet_tlaku(long d1,long d2){
    long ut20,off,sens,p,dut;
    ut20 = 8*C[4]+10000;
    dut = d2-ut20;
    off = C[1]+(C[3]-250)*dut/4096+10000;
    sens = C[0]/2+(C[2]+200)*dut/8192+3000;
```

```
p = sens * (d1-off)/2048+1000; //zase spatne 2^11 != 2046 ani 4096 to zpusobuje chybu o
velikosti 5 na vysledku, je rovno 2048!!
```

```
    return(p);
}
long vypocet_teploty(long d2)
{
    long temp,dut,ut20;
    ut20 = 8*C[4]+10000;
    dut = d2-ut20;
    temp = 200+dut*(C[5]+100)/2048;

    return(temp);
}
```

```
void posli_po_SPI_od_nejnizsiho_bitu(UINT8 data_k_poslani, UINT8 pocet_dat_k_poslani)
//SerialSendLsbFirst
```

```
{
    UINT8 i;
    UINT8 c;
    LATBbits.LATB9 = 0;    // hodiny do 0 - pro jistotu jako zacatek komunikace
    mikro_delay();
    for (i=0; i<pocet_dat_k_poslani; i++)
    {
        c = (data_k_poslani & 1);
        if (c==1)
        {
            LATBbits.LATB11 = 1;    // najnizsi bit je 1, poslu 1
        }
        else
        {
            LATBbits.LATB11 = 0;    // najnizsi bit je 0, poslu 0
        }
        LATBbits.LATB9 = 1;
        mikro_delay();
        LATBbits.LATB9 = 0;
        data_k_poslani = (UINT8) (data_k_poslani >> 1);    // vstup o jedno doprava -> zahodi se
        najnizsi bit
    }
}

long nacti_a_prepocitej_tlak(void)
{
    long D1,D2;
    UINT8 cislo;
    D1 = dostan_D1();
    D2 = dostan_D2();
}
```

```

for(cislo=0; cislo<6; cislo++)
{
    C[cislo] = Prevod_Wordu(cislo+1,word1,word2,word3,word4);
}
return(vypocet_tlaku(D1,D2));
}
long nacti_a_prepocitej_teplotu(void)
{ long D2;
  UINT8 cislo;
  D2 = dostan_D2();
for(cislo=0; cislo<6; cislo++)
{
    C[cislo] = Prevod_Wordu(cislo+1,word1,word2,word3,word4);
}
return(vypocet_teploty(D2));
}
/*
=====
= */
/* reset_senzoru */
/* ===== */
void reset_senzoru(void)
{
    posli_po_SPI_od_nejnizsiho_bitu(0x55, 8); // 0x55 = 01010101 => posle se po spi 10101010
    posli_po_SPI_od_nejnizsiho_bitu(0x55, 8); // 0x55 = 01010101 => posle se po spi 10101010
    posli_po_SPI_od_nejnizsiho_bitu(0x00, 5); // 0x00 = 00000 => posle se po spi 00000
}
/*
=====
= */
/* vycti_16_bitu_ze_senzoru */
/*
=====
= */
UINT16 vycti_16_bitu_ze_senzoru(void)
{
    UINT8 i;
    UINT16 v;
    v = 0;
    LATBbits.LATB9 = 0; // hodiny do 0 - pro jistotu jako zacatek komunikace
    mikro_delay();
    for (i=0; i<16; i++)
    {
        LATBbits.LATB9 = 1; // nastupna hrana hodin
    }
}

```

```

    mikro_delay();
    LATBbits.LATB9 = 0; // sestupna hrana hodin -> ted ma senzor pripravena data
    v = v << 1; // uvolnim si najnizsi bit prijateho wordu pro aktualni bit
    if (PORTBbits.RB10 == 1) // pokud je na vstupnim pinu DO jednicka
    {
        v = v | 1; // nastavim jednicku na najnizsi bit, jinak tam zustane nula
    }
    mikro_delay();
}
return(v);
}

/*
=====

= */
/* ziskej_konfiguracni_word */
/*
=====

= */
long ziskej_konfiguracni_word (UINT8 index_konfiguracniho_wordu) // 1 az 4
{
    long data = 0;
    switch(index_konfiguracniho_wordu)
    {
        case 1:
            posli_po_SPI_od_najnizsiho_bitu(0x57, 8); // posle se 11101010
            posli_po_SPI_od_najnizsiho_bitu(0x01, 5); // posle se 10000
            data = vycti_16_bitu_ze_senzoru();
            break;
        case 2:
            posli_po_SPI_od_najnizsiho_bitu(0xD7, 8); // posle se 11101011
            posli_po_SPI_od_najnizsiho_bitu(0x00, 5); // posle se 00000
            data = vycti_16_bitu_ze_senzoru();
            break;
        case 3:
            posli_po_SPI_od_najnizsiho_bitu(0x37, 8); // posle se 11101100
            posli_po_SPI_od_najnizsiho_bitu(0x01, 5); // posle se 10000
            data = vycti_16_bitu_ze_senzoru();
            break;
        case 4:
            posli_po_SPI_od_najnizsiho_bitu(0xB7, 8); // posle se 11101101
            posli_po_SPI_od_najnizsiho_bitu(0x00, 5); // posle se 00000
            data = vycti_16_bitu_ze_senzoru();
            break;
    }
}

```



```

    }
    posli_po_SPI_od_nejnizsiho_bitu(0x00, 1); // zaverecna 0 dle datasheetu
    return(data);
}
/*
=====

= */
/* posli_po_SPI_od_nejnizsiho_bitu */
/*
=====

= */
long dostan_D1 (void)
{
    long d1;
    posli_po_SPI_od_nejnizsiho_bitu(0x2F, 8); //posle po RB11 11110100 setup bit
    posli_po_SPI_od_nejnizsiho_bitu(0x00, 2); //posle stop bit 00
    while(PORTBbits.RB10 == 0); //ceka dokud na DI nebude 1
    posli_po_SPI_od_nejnizsiho_bitu(0x00, 2); //nastane DI 1 bude a posle dva hodinova bity
    while(PORTBbits.RB10 == 1); //ceka dokud na DI nebude 0
    d1 = vyci_16_bitu_ze_senzoru();
    posli_po_SPI_od_nejnizsiho_bitu(0x00, 1); //posle hodinovy impuls potrebný pro ukončení
    komunikace
    return(d1);
}
long dostan_D2 (void)
{
    long d2;
    posli_po_SPI_od_nejnizsiho_bitu(0x4F, 8);
    posli_po_SPI_od_nejnizsiho_bitu(0x00, 3);
    while(PORTBbits.RB10 == 0); //ceka dokud na DI nebude 1
    posli_po_SPI_od_nejnizsiho_bitu(0x00, 1);
    while(PORTBbits.RB10 == 1);
    d2 = vyci_16_bitu_ze_senzoru();
    posli_po_SPI_od_nejnizsiho_bitu(0x00, 1);
    return(d2);
}

/* ----- */
/* ----- Prevod_WORDU ----- */
/* ----- */
long Prevod_Wordu (UINT8 cislo, long W1, long W2, long W3, long W4)
{
    long c;
    long x, y;

```

```

c = 0;
switch(cislo)
{
    case 1:
        c = (W1 >> 3) & 0x1FFF;
        break;

    case 2:
        x = (W1 <<10) & 0x1C00;
        y = (W2 >> 6) & 0x03FF;
        c = x | y;
        break;

    case 3:
        c = (W3 >> 6) & 0x03FF;
        break;

    case 4:
        c = (W4 >> 7) & 0x01FF;
        break;

    case 5:
        x = (W2 << 6 )& 0x0FC0;
        y = W3      & 0x003F;
        c = x | y;
        break;

    case 6:
        c = W4 & 0x007F;
        break;

}
return(c);
}

void delay(void)
{UINT8 a;
  UINT8 b;
    for (a=0;a<254;a++)
    {
        a++;
        a--;
        for (b=0;b<254;b++)
        {
            b++;
            b--;
        }
    }
}

void mikro_delay(void){

```

```

    UINT8 i;
    for(i=0;i<5;i++) //3 cykly pro 2uS
    {
        i++;
        i--;
    }
}

void Inicializace_senzoru(void){
    reset_senzoru();
    word1 = ziskej_konfiguracni_word(1);
    word2 = ziskej_konfiguracni_word(2);
    word3 = ziskej_konfiguracni_word(3);
    word4 = ziskej_konfiguracni_word(4);

}

```

Příloha II.

```
/*header soubor pro funkce senzoru*/  
long vypocet_tlaku(long,long);  
long vypocet_teploty(long);  
void posli_po_SPI_od_nejnizsiho_bitu(UINT8, UINT8);  
long nacti_a_prepocitej_tlak();  
long nacti_a_prepocitej_teplotu();  
void reset_senzoru(void);  
UINT16 vyci_16_bitu_ze_senzoru(void);  
long ziskej_konfiguracni_word (UINT8);  
long dostan_D1 (void);  
long dostan_D2 (void);  
long Prevod_Wordu (UINT8, long, long, long, long);  
void delay(void);  
void mikro_delay(void);  
void Inicializace_senzoru(void);  
UINT32 get_pulsy(void);  
void vynulovani_pulsu(void);
```

Příloha III.

```
/*Inicializace LED, Timeru, Interruptu pomoci .h souboru pro blikani LED*/
#define REGISTR_PR1()    PR1 = 0xFA; //vyssi frekvenci, musis mu orezat, at netika od 0000
do FFFF ale treba jen do 0x1000 -> tim to 16x zrychlis
#define MC_INIC()        TRISBbits.TRISB2 = 0; LATBbits.LATB2 = 0;
#define MC_TOGGLE()      LATBbits.LATB2 = ~LATBbits.LATB2;
#define LED_TOGGLE()     LATBbits.LATB7 = ~LATBbits.LATB7;
//#define LED_ZAP()       LATBbits.LATB7 = 1;
//#define LED_VYP()       LATBbits.LATB7 = 0;
#define T1_ZAP           T1CONbits.TON = 1;
#define T1_PRESSEDDEL_1  T1CONbits.TCKPS = 0;
#define T1_VNITR_OSC     T1CONbits.TCS = 0; //T1CON = 0xfffd; // Druhy zapis definice
pomoci registru a hex tvaru. Ktery bitovy zapisem urci zapnuti urcite funkce
#define T1_INTR_ZAP      IEC0bits.T1IE = 1;
#define T1_RESET_INTR_FLAG IFS0bits.T1IF = 0;

/*Piny pro konfiguraci SPI softwarove*/
#define SPI2_INIC          TRISBbits.TRISB9 = 0; TRISBbits.TRISB11 = 0;
TRISBbits.TRISB10 = 1;
#define SPI2_DI            //RB10
#define SPI2_DO            //RB11
#define SPI2_SCK_OUT       //RB9
void Inicializace_T1(void);
```

Příloha IV.

```
#define SOUBOR          "senzor.bin"
/*nastaveni komunikace senzoru*/
SPI2_INIC;
long real_tlak, real_teplo = 0;
UINT16 mojeData[8] = {real_tlak, real_teplo};
FSFILE *ptSoubor;
UINT32 nactene_pulzy=0;
Inicializace_T1();
Inicializace_senzoru();
ptFile = FSfopen(FILENAME2, WRITE);
ptSoubor = FSfopen(SOUBOR,WRITE); // využití předchozí funkce pro vytvoření souboru pro
ukládání
if(ptFile == NULL || ptSoubor == NULL){
    ioSetLed(TRUE);
    while(1);
}
nactene_pulzy = get_pulsy();
if (nactene_pulzy > 32000){
    reset_senzoru();
    real_tlak = nacti_a_prepocitej_tlak();
    real_teplo = nacti_a_prepocitej_teplo();
    /*Uložení dat na SD kartu*/

    mojeData[0] = real_tlak;
    mojeData[1] = real_teplo;
    mojeData[2]=0;
    mojeData[3]=0;
    mojeData[4]=0;
    mojeData[5]=0;
    mojeData[6]=0;
    mojeData[7]=0;
    FSfwrite((void*) mojeData, 16, 1, ptSoubor);
    vynulovani_pulsu();

}
```

Příloha V.

```
function varargout = Data_senzor(varargin)
% DATA_SENZOR M-file for Data_senzor.fig
%     DATA_SENZOR, by itself, creates a new DATA_SENZOR or raises the
existing
%     singleton*.
%
%     H = DATA_SENZOR returns the handle to a new DATA_SENZOR or the
handle to
%     the existing singleton*.
%
%     DATA_SENZOR('CALLBACK',hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in DATA_SENZOR.M with the given input
arguments.
%
%     DATA_SENZOR('Property','Value',...) creates a new DATA_SENZOR
or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before Data_senzor_OpeningFcn gets called.
An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to Data_senzor_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Data_senzor

% Last Modified by GUIDE v2.5 25-Apr-2012 15:11:13

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Data_senzor_OpeningFcn, ...
                  'gui_OutputFcn',  @Data_senzor_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```

% --- Executes just before Data_senzor is made visible.
function Data_senzor_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Data_senzor (see VARARGIN)
% Choose default command line output for Data_senzor
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes Data_senzor wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Data_senzor_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in nacti.
function nacti_Callback(hObject, eventdata, handles)
% hObject    handle to nacti (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[FileName, PathName] = uigetfile('*.bin','otevri se')
if(FileName)~=0
data = fullfile([PathName, FileName]);
    fid = fopen(data);
    hodnoty = fread(fid,'uint16','1');
else
    warndlg('Nezvoleny hodnoty','Chyba')
end
tlak=hodnoty(1:8:end);
metry=tlak*0.01;
temp=hodnoty(2:8:end);
teplota=temp/10;
if (tlak)~=0
size(hodnoty);
i=1;
k=0;
axes(handles.axes1)
for i=1:1:(size(tlak))
    pocitadlo(i,1)=k;
    k=k+0.65;
end
[AH,H1,H2]=plotyy(pocitadlo,tlak,pocitadlo,metry);
set(get(AH(1),'Ylabel'),'String','Hydrostatický tlak [mbar]')
set(get(AH(2),'Ylabel'),'String','Hloubka [m]')
xlabel('t [s]');
grid on;
hold on;

```



```

axes(handles.axes2)
y=1;
t=0;
for y=1:1:(size(teplota))
    count(y,1)=t;
    t=t+0.65;
end
plot(count,teplota,'r');
ylabel('Teplota [°C]');
xlabel('t[s]');
grid on;
hold on;
else
    warndlg('Nulová hodnota dat','Chyba')

end
% --- Executes on button press in zavrit.
function zavrit_Callback(hObject, eventdata, handles)
% hObject    handle to zavrit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
clear all
close all

% --- Executes during object creation, after setting all properties.
function axes4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
% Hint: place code in OpeningFcn to populate axes4
axes(hObject)
imshow('logo.png')

```